

RH850/P1M-E Group

User's Manual: Hardware

Renesas microcontroller
RH850 Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

Notes for CMOS devices

- (1) Voltage application waveform at input pin:** Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).
- (2) Handling of unused input pins:** Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
- (3) Precaution against ESD:** A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.
- (4) Status before initialization:** Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.
- (5) Power ON/OFF sequence:** In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.
- (6) Input of signal during power off state:** Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers This manual is intended for users who wish to understand the functions of the RH850/P1M-E and design application systems using the following RH850/P1M-E microcontrollers:

Purpose This manual is intended to give users an understanding of the hardware functions of the RH850/P1M-E shown in the *Organization* below.

Organization This manual is divided into two parts: Hardware (this manual) and Architecture (RH850G3M User's Manual: Software).

| Hardware | Software |
|------------------------------|---------------------------|
| Pin functions | Overview |
| CPU function | Processor Model |
| On-chip peripheral functions | Register Reference |
| Flash memory programming | Exceptions and Interrupts |
| | Memory Management |
| | Instruction Reference |
| | Reset |
| | Appendix |

How to read this manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To understand the overall functions of the RH850/P1M-E.

→ Read this manual according to the Contents.

To understand the details of an instruction function

→ See RH850G3M User's Manual: Software (R01US0123E) available separately.

Conventions Data significance: Higher digits on the left and lower digits on the right
Active low representation: xxx (overscore over pin or signal name)
Memory map address: Higher addresses on the top and lower addresses on the bottom
Note: Footnote for item marked with Note in the text
Caution: Information requiring particular attention
Remark: Supplementary information
Numeric representation: Binary ... xxxx or xxxx_B
Decimal ... xxxx
Hexadecimal ... xxxx_H
Prefix indicating power of 2 (address space, memory capacity):
K (kilo): $2^{10} = 1,024$
M (mega): $2^{20} = 1,024^2$
G (giga): $2^{30} = 1,024^3$

Description of Registers

Each register description includes register access, register address, and register value after a reset, a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meaning of the bit settings.

The standard format for bit charts and tables are described below.

Table 14.19 CSIGNCFG0 register contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | |
|--------------|----------------|---|--|----------|--------------|-----------|---|---|-----------------------|--------------------------|---|---|---------------------------|--|---|---|----------------|-------------------------------|---|---|-----------------|--------------------------------|
| 31, 30 | Reserved | The write value should always be the value after reset. | | | | | | | | | | | | | | | | | | | | |
| 29, 28 | CSIGNPS[1:0] | Specifies parity. <table border="1"> <thead> <tr> <th>CSIGNPS1</th> <th>CSIGNPS0</th> <th>Transmission</th> <th>Reception</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No parity transmitted</td> <td>No parity is waited for.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Add parity bit fixed at 0</td> <td>Parity bit is waited for but not judged.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Add odd parity</td> <td>Odd parity bit is waited for.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Add even parity</td> <td>Even parity bit is waited for.</td> </tr> </tbody> </table> | CSIGNPS1 | CSIGNPS0 | Transmission | Reception | 0 | 0 | No parity transmitted | No parity is waited for. | 0 | 1 | Add parity bit fixed at 0 | Parity bit is waited for but not judged. | 1 | 0 | Add odd parity | Odd parity bit is waited for. | 1 | 1 | Add even parity | Even parity bit is waited for. |
| CSIGNPS1 | CSIGNPS0 | Transmission | Reception | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | No parity transmitted | No parity is waited for. | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Add parity bit fixed at 0 | Parity bit is waited for but not judged. | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Add odd parity | Odd parity bit is waited for. | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Add even parity | Even parity bit is waited for. | | | | | | | | | | | | | | | | | | | |
| 27 to 24 | CSIGNDLS [3:0] | Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits CAUTION For a data length of less than 7 bits, do not set bits CSIGNCFG0.CSIGNDLS[3:0] for a value 1 to 6 when the extended data length function is disabled with bit CSIGNCTL1.CSIGNEDLE set to 0. It is forbidden to transmit two consecutive data with a data length of less than 7 bits. | | | | | | | | | | | | | | | | | | | | |
| 23 to 19 | Reserved | The write value should always be the value after reset. | | | | | | | | | | | | | | | | | | | | |

(1) Access

The register can be accessed in the bit unit indicated here.

(2) Address

This is the register address.

For base address, see description of base address in each section.

(3) Value after a reset (in hexadecimal notation)

This is the value of all bits of the register after a reset. Values for bytes are given as numbers in the range from 0 to 9 and letters from A to F or as X where they are undefined.

(4) Bit position

This is the bit number.

The bits are numbered from 31 to 0 for 32-bit registers, 15 to 0 for 16-bit registers, and 7 to 0 for 8-bit registers.

(5) Bit name

Bit name or field name is indicated.

When clearly identifying the digits of a bit field is required, do so by using a form such as CSIGNDLS[3:0] above.

Indicate reserved bits by using a dash (—).

(6) Value after a reset (in binary notation)

This is the bit values after a reset.

0 : The value after a reset is 0.

1 : The value after a reset is 1.

— : The value after a reset is undefined.

(7) R/W

This is the bit attribute of all bits of the register.

R/W : The bit or field is readable and writable.

R : The bit or field is readable.

Note that all reserved bits are indicated as R.

When written, the value specified in the bit chart or the value after a reset should be written.

In case of writing to writable registers that also include non-reserved bits with the R-attribute, writing to the R-attribute bits will be ignored unless otherwise specified.

W : This bit or field is writable. When read, the value is undefined. If a value is indicated in the bit chart, the value is returned.

(8) Function

This is function of the bit.

Table of Contents

| | | |
|-----------|--|-----|
| Section 1 | Overview | 64 |
| 1.1 | Outline | 64 |
| 1.2 | Application Fields..... | 65 |
| 1.3 | Specification Overview | 65 |
| 1.4 | Block Configuration..... | 67 |
| Section 2 | Pin Functions | 68 |
| 2.1 | Pin Connection Diagrams | 68 |
| 2.2 | Pin Expansion..... | 75 |
| 2.2.1 | Overview..... | 75 |
| 2.2.2 | List of Pin Functions | 75 |
| 2.3 | Port Functions..... | 91 |
| 2.3.1 | Features | 91 |
| 2.3.2 | Overview..... | 92 |
| 2.3.2.1 | Terms | 92 |
| 2.3.2.2 | Overview of Pin Functions..... | 93 |
| 2.3.2.3 | Pin Data Input/Output..... | 95 |
| 2.3.3 | Port Type..... | 97 |
| 2.3.4 | Port Group Configuration Register | 99 |
| 2.3.4.1 | Outline | 99 |
| 2.3.4.2 | Pin Function Configuration | 101 |
| 2.3.4.3 | Pin Data Input/Output..... | 111 |
| 2.3.4.4 | Configuration of Electrical Characteristics..... | 117 |
| 2.3.4.5 | Pin-unit Register..... | 124 |
| 2.3.4.6 | Example of Port Configuration Flowchart..... | 126 |
| 2.3.5 | Functional Selection | 131 |
| 2.3.5.1 | Register Configuration in Use of the Alternative Function..... | 131 |
| 2.3.5.2 | Alternative Function to be used in Direct I/O Control Alternative Mode | 131 |
| 2.3.5.3 | Register Setting in Use of an Analog Input Pin | 132 |
| 2.3.5.4 | Input Buffer Control (PISA)..... | 132 |
| 2.3.5.5 | Output Buffer Control (PDSC, PUCC)..... | 134 |
| 2.4 | Organization of Port Groups | 137 |
| 2.4.1 | Port Function | 137 |
| 2.4.1.1 | List of Port Registers | 137 |
| 2.4.1.2 | List of Alternative Function Pins | 149 |
| 2.4.1.3 | Port 0 (P0)..... | 149 |
| 2.4.1.4 | Port 1 (P1)..... | 150 |
| 2.4.1.5 | Port 2 (P2)..... | 151 |
| 2.4.1.6 | Port 3 (P3)..... | 152 |
| 2.4.1.7 | Port 4 (P4)..... | 153 |
| 2.4.1.8 | Port 5 (P5)..... | 154 |
| 2.4.1.9 | Port JP0 (JP0)..... | 155 |
| 2.5 | DNF | 156 |
| 2.5.1 | Example of Noise Elimination..... | 156 |
| 2.6 | Noise Filter and Edge Level Detection Circuit | 157 |
| 2.6.1 | Allocation of Port Filters..... | 157 |

| | | |
|-----------------------------------|---|------------|
| 2.6.2 | Filter Type..... | 164 |
| 2.6.2.1 | Digital Filter Type A (INTP/RXD Alternative) Input Pin..... | 164 |
| 2.6.2.2 | Digital Filter Type B (INTP/ESO Alternative) Input Pin..... | 165 |
| 2.6.2.3 | Digital Filter Type C (for INTP9 and NMI) Input Pin | 166 |
| 2.6.2.4 | Digital Filter Type D (for SENT, PSI5) Input Pin..... | 167 |
| 2.6.2.5 | Digital Filter Type E (No Edge Detection) Input Pin | 168 |
| 2.6.2.6 | Digital Filter Type F (No Edge Detection) Input Pin | 168 |
| 2.6.2.7 | Analog Filter ANF Input Pin..... | 169 |
| 2.6.3 | Registers | 170 |
| 2.6.3.1 | List of Registers..... | 170 |
| 2.6.3.2 | FCLAnCTLm — Filter Control Register..... | 171 |
| 2.6.3.3 | DNFAnCTL — Digital Noise Elimination Control Register | 172 |
| 2.6.3.4 | DNFAnEN — Digital Noise Elimination Enable Register..... | 173 |
| 2.6.3.5 | DNFAnENL — Digital Noise Elimination Enable Register..... | 173 |
| 2.6.3.6 | DNFCKSnC — Digital Noise Elimination Sampling Clock Source Selection Register..... | 174 |
| 2.6.3.7 | DNFCSCnSTAT — Digital Noise Elimination Sampling Clock Source Status Register | 175 |
| 2.6.4 | Notes for Digital Noise Filter..... | 176 |
| 2.7 | Pin State | 177 |
| 2.8 | Handling of Unused Pins | 180 |
| Section 3 CPU System | | 187 |
| 3.1 | Overview..... | 187 |
| 3.1.1 | Block Configuration | 187 |
| 3.2 | CPU | 189 |
| 3.2.1 | Core Functions | 189 |
| 3.2.1.1 | Features | 189 |
| 3.2.1.2 | Register Set..... | 190 |
| 3.2.2 | Instruction Cache and Data Buffer | 227 |
| 3.2.2.1 | Features | 227 |
| 3.2.2.2 | Instruction Cache Function..... | 228 |
| 3.2.2.3 | Data Buffer Function | 229 |
| 3.2.3 | Reliability Functions..... | 230 |
| 3.2.3.1 | PE Guard Function (PEG)..... | 230 |
| 3.2.3.2 | PE's Internal Peripherals Protection Function (IPG) | 237 |
| 3.2.3.3 | System Error Notification Control Function (SEG) | 244 |
| 3.2.3.4 | Checker Core | 250 |
| 3.3 | Inter CPU Functions Overview | 251 |
| 3.3.1 | Processor Element Identifier | 251 |
| 3.3.2 | Exclusive Function..... | 251 |
| 3.3.3 | Write-Through Buffer for Global RAM | 251 |
| 3.4 | Usage Notes | 254 |
| 3.4.1 | Synchronization of Store Instruction Completion and Subsequent Instruction Generation..... | 254 |
| 3.4.1.1 | When updated results in the control registers are reflected in the implementation of a subsequent instruction: | 254 |

| | | |
|------------------|--|------------|
| 3.4.1.2 | When the updated results of the control registers or memory to be used in the instruction fetch of the subsequent instruction: | 255 |
| 3.4.1.3 | When switching the code flash memory area:..... | 255 |
| 3.4.2 | Accesses to Registers by Bit-Manipulation Instructions | 255 |
| 3.4.3 | Ensuring Coherency after Code Flash Programming..... | 255 |
| 3.4.4 | Overwriting Context when Acknowledging Multiple Exceptions | 255 |
| 3.4.5 | Usage Notes on Prefetching..... | 256 |
| 3.4.6 | Usage Note when Exception is Acknowledged | 256 |
| Section 4 | Address Space..... | 257 |
| 4.1 | Address Space | 257 |
| 4.2 | Address Space Viewed from Each Bus Master | 258 |
| 4.2.1 | Space in which instructions can be fetched..... | 258 |
| 4.2.2 | Data space accessible by PE1 | 258 |
| 4.2.3 | Data space accessible by DMA (DMAC, DTS)..... | 258 |
| 4.2.4 | Data space accessible by H-Bus master | 259 |
| 4.3 | Guard Function | 260 |
| Section 5 | Operating Modes | 261 |
| 5.1 | Features..... | 261 |
| 5.1.1 | Normal Operating Mode | 261 |
| 5.1.2 | Serial Programming mode..... | 261 |
| 5.2 | External Input Pins..... | 262 |
| 5.3 | Register Description | 263 |
| 5.3.1 | List of Registers..... | 263 |
| 5.3.2 | MODE — Mode Register..... | 263 |
| Section 6 | Interrupt..... | 264 |
| 6.1 | Overview..... | 264 |
| 6.1.1 | Reset Sources | 264 |
| 6.2 | Register Specifications | 265 |
| 6.2.1 | Register Configuration..... | 265 |
| 6.2.2 | EIC0 - EIC383 — EI Level Interrupt Control Registers 0 to 383..... | 267 |
| 6.2.3 | IMR0 - IMR11 — EI Level Interrupt Mask Registers 0 to 11 | 269 |
| 6.2.4 | EIBD0 - EIBD383 — EI Level Interrupt Bind Registers 0 to 383 | 271 |
| 6.2.5 | FNC — FE Level NMI Control Register..... | 272 |
| 6.2.6 | FIC — FE Level Interrupt Control Register..... | 272 |
| 6.2.7 | SINTR0 - SINTR4 — Software Interrupt Registers | 273 |
| 6.2.8 | PINT0 — PINT7 — Peripheral Interrupt Status Registers PINTCLR0 - PINTCLR7 — Peripheral Interrupt Status Clear Registers | 274 |
| 6.2.9 | FEINTF — FEINT Factor Register | 278 |
| 6.2.10 | FEINTFC — FEINT Factor Clear Register | 279 |
| 6.3 | Interrupt Sources | 280 |
| 6.3.1 | NMI Interrupts..... | 280 |
| 6.3.2 | INTPn Interrupts | 280 |

| | | |
|------------------|---|------------|
| 6.3.3 | ECM Interrupts | 280 |
| 6.3.4 | Software Interrupts | 280 |
| 6.3.5 | On-Chip Peripheral Module Interrupts..... | 280 |
| 6.4 | Interrupt Exception Handler and Priority Operations | 281 |
| 6.5 | Operation | 291 |
| 6.5.1 | External Interrupts (NMI / INTP) | 291 |
| 6.5.2 | Software Interrupt | 291 |
| 6.5.3 | DTS Interrupt Merge Function | 291 |
| 6.5.4 | Interrupt Processing Flow..... | 292 |
| 6.5.4.1 | NMI Processing Flow | 292 |
| 6.5.4.2 | External Interrupt Processing Flow | 293 |
| 6.5.4.3 | Software Interrupt Processing Flow | 294 |
| 6.5.4.4 | DTS Interrupt Processing Flow | 295 |
| 6.6 | Interrupt latency | 297 |
| 6.7 | Using Interrupt Request Signals to Initiate Data Transfer | 297 |
| Section 7 | DMA | 298 |
| 7.1 | Features of RH850/P1M-E DMA | 298 |
| 7.1.1 | Number of Channels..... | 298 |
| 7.1.2 | Register Base Address..... | 299 |
| 7.1.3 | Interrupt Requests | 299 |
| 7.1.4 | DMA Trigger Source..... | 300 |
| 7.1.5 | DTS Trigger Source..... | 304 |
| 7.2 | Overview..... | 308 |
| 7.2.1 | Overview..... | 308 |
| 7.2.2 | Term Definition | 308 |
| 7.3 | DMA Function | 309 |
| 7.3.1 | Basic Operation of DMA Transfer..... | 309 |
| 7.3.1.1 | Transfer Mode..... | 309 |
| 7.3.1.2 | Executing a DMA Cycle..... | 309 |
| 7.3.1.3 | Updating Transfer Information..... | 309 |
| 7.3.1.4 | Last Transfer and Address Reload Transfer | 310 |
| 7.3.1.5 | Transfer Completion Interrupt and Transfer Count Match Interrupt Outputs..... | 310 |
| 7.3.1.6 | Continuous Transfer..... | 311 |
| 7.3.2 | Channel Priority Order..... | 313 |
| 7.3.2.1 | DMAC Channel Arbitration | 313 |
| 7.3.2.2 | DTS Channel Arbitration | 314 |
| 7.3.2.3 | Interface Arbitration | 315 |
| 7.3.3 | Reload Function | 316 |
| 7.3.3.1 | Overview of the Reload Function | 316 |
| 7.3.3.2 | Operation of Reload Function 1 | 316 |
| 7.3.3.3 | Reload Function 2 | 317 |
| 7.3.3.4 | Timing of Setting DMAC Reload Registers | 319 |
| 7.3.3.5 | Timing of Setting DTS Reload Registers..... | 319 |
| 7.3.4 | Chain Function | 320 |
| 7.3.4.1 | Overview | 320 |

| | | |
|---------|---|-----|
| 7.3.4.2 | Setting Up the Chain Function | 321 |
| 7.3.4.3 | Caution for Using the Chain Function | 321 |
| 7.3.5 | DMAC Operation | 322 |
| 7.3.5.1 | Types of DMA Transfer Requests and Assigning DMA Transfer Requests..... | 322 |
| 7.3.5.2 | Generating and Accepting a Hardware DMA Transfer Request | 322 |
| 7.3.5.3 | Generating and Accepting a Software DMA Transfer Request..... | 324 |
| 7.3.5.4 | Execution of DMA transfer | 325 |
| 7.3.6 | DTS Operation..... | 328 |
| 7.3.6.1 | Types of DMA Transfer Requests and Assigning DMA Transfer Requests..... | 328 |
| 7.3.6.2 | Generating and Accepting a DMA Transfer Request | 328 |
| 7.3.6.3 | Executing DMA Transfer | 328 |
| 7.3.6.4 | DTSRAM Access..... | 330 |
| 7.3.6.5 | Execution Time of DTS Transfers | 330 |
| 7.4 | Temporarily Suspending DMA Transfers..... | 331 |
| 7.4.1 | Suspension, Restart and Abortion of a DMA Channel | 331 |
| 7.4.2 | Suspension, Resume, and Transfer Abort of a DTS | 332 |
| 7.4.3 | Masking and Clearing a Hardware DMA Transfer Request by the DTFR | 333 |
| 7.4.4 | Masking and Clearing a Hardware DMA Transfer Request by the DTSFSL..... | 333 |
| 7.4.5 | List of Suspend, Resume, and Transfer Abort Functions..... | 333 |
| 7.5 | Error Control | 334 |
| 7.5.1 | Type of Error..... | 334 |
| 7.5.2 | DMA Transfer Error | 334 |
| 7.5.2.1 | Operation of a DMAC When DMA Transfer Error Occurs..... | 334 |
| 7.5.2.2 | Operation of a DTS When DMA Transfer Error Occurs | 334 |
| 7.5.3 | DTSRAM Error | 335 |
| 7.5.3.1 | DTSRAM ECC Test..... | 335 |
| 7.5.3.2 | Stimulation of DTSRAM ECC Error..... | 336 |
| 7.6 | Reliability Function..... | 337 |
| 7.6.1 | Overview..... | 337 |
| 7.6.2 | Register Access Protection Function..... | 337 |
| 7.6.2.1 | Identifying the Accessing Master..... | 337 |
| 7.6.2.2 | Special Master Access | 337 |
| 7.6.2.3 | General Master Access | 337 |
| 7.6.2.4 | Channel Assignment | 337 |
| 7.6.2.5 | Illegal Access | 338 |
| 7.6.3 | Master Information Inherit Function..... | 339 |
| 7.6.4 | Other Reliability Functions..... | 339 |
| 7.6.4.1 | Restriction on the Next Channel in the Chain | 339 |
| 7.7 | Setting Up DMA Transfer..... | 340 |
| 7.7.1 | Overview of Setting Up DMA..... | 340 |
| 7.7.2 | Setting Up the Overall DMA Operation..... | 341 |
| 7.7.3 | Setting Up the DMA Channel Setting | 342 |
| 7.7.3.1 | Setting Up the DMAC Channel Setting | 342 |
| 7.7.3.2 | Setting Up the DTS Channel Setting..... | 343 |
| 7.8 | Global Register | 344 |
| 7.8.1 | List of Global Register Address | 344 |

| | | |
|----------|---|-----|
| 7.8.2 | Details of Global Registers | 346 |
| 7.8.2.1 | DMACTL — DMA Control Register | 346 |
| 7.8.2.2 | DTSCTL1 — DTS Control Register 1 | 347 |
| 7.8.2.3 | DTSCTL2 — DTS Control Register 2 | 348 |
| 7.8.2.4 | DTSSTS — DTS Status Register | 349 |
| 7.8.2.5 | DMACER — DMAC Error Register | 350 |
| 7.8.2.6 | DTSER1 — DTS Error Register 1 | 351 |
| 7.8.2.7 | DTSER2 — DTS Error Register 2 | 352 |
| 7.8.2.8 | DTSERC — DTS Error Clear Register | 353 |
| 7.8.2.9 | DM0CMV — DMAC0 Register Access Protection Violation Register | 354 |
| 7.8.2.10 | DM1CMV — DMAC1 Register Access Protection Violation Register | 355 |
| 7.8.2.11 | DTSCMV — DTS Register Access Protection Violation Register | 356 |
| 7.8.2.12 | CMVC — Register Access Protection Violation Clear Register | 357 |
| 7.8.2.13 | Transfer Status Register (TFRSTS) | 358 |
| 7.8.2.14 | DTSPRy — DTS Channel Priority Setting (y = 0 to 7) | 359 |
| 7.8.2.15 | DTRECCTL — DTSRAM ECC Control Register | 363 |
| 7.8.2.16 | DTRERINT — DTSRAM Error Notification Control Register | 364 |
| 7.8.2.17 | DTRTCTL — DTSRAM Test Control Register | 365 |
| 7.8.2.18 | DTRTWDAT — DTSRAM Test Write Data Register | 366 |
| 7.8.2.19 | DTRTRDAT — DTSRAM Test Read Data Register | 367 |
| 7.8.2.20 | ADECCTCL — ECC on BUS Address ECC Test Control Register | 368 |
| 7.8.2.21 | ADECCTDT — ECC on BUS Address ECC Test Data Register | 369 |
| 7.8.2.22 | DMxCM — DMAC Channel Master Setting (x = 00 to 07, 10 to 17) | 370 |
| 7.8.2.23 | DTSmCM — DTS Channel Master Setting Register (m = 000 to 127) | 371 |
| 7.9 | DMAC Channel Register | 373 |
| 7.9.1 | DMAC Channel Register Address | 373 |
| 7.9.2 | Details of DMAC Channel Registers | 374 |
| 7.9.2.1 | DSAn — DMAC Source Address Register | 374 |
| 7.9.2.2 | DDAn — DMAC Destination Address Register | 375 |
| 7.9.2.3 | DTCn — DMAC Transfer Count Register | 376 |
| 7.9.2.4 | DTCTn — DMAC Transfer Control Register | 377 |
| 7.9.2.5 | DRSAn — DMAC Reload Source Address Register | 380 |
| 7.9.2.6 | DRDAn — DMAC Reload Destination Address Register | 381 |
| 7.9.2.7 | DRTCn — DMAC Reload Transfer Count Register | 382 |
| 7.9.2.8 | DTCCn — DMAC Transfer Count Compare Register | 383 |
| 7.9.2.9 | DCENn — DMAC Channel Operation Enable Setting Register | 384 |
| 7.9.2.10 | DCSTn — DMAC Transfer Status Register | 385 |
| 7.9.2.11 | DCSTSn — DMAC Transfer Status Set Register | 387 |
| 7.9.2.12 | DCSTCn — DMAC Transfer Status Clear Register | 388 |
| 7.9.2.13 | DTFRn — DTFR Setting Register | 389 |
| 7.9.2.14 | DTFRRQn — DTFR Transfer Request Status Register | 390 |
| 7.9.2.15 | DTFRRQCn — DTFR Transfer Request Clear Register | 391 |
| 7.10 | DTS Channel Register | 392 |
| 7.10.1 | Transfer information of the DTS (TI) | 392 |
| 7.10.1.1 | Structure of the TI | 392 |
| 7.10.1.2 | Organization of the TI in the DTSRAM | 393 |
| 7.10.1.3 | Accessing the TI | 394 |
| 7.10.1.4 | Caution about Accessing the TI | 394 |
| 7.10.2 | DTS Channel Register Address | 395 |

| | | |
|------------------|--|------------|
| 7.10.3 | Details of DTS Channel Registers..... | 396 |
| 7.10.3.1 | DTSAm — DTS Source Address Register..... | 396 |
| 7.10.3.2 | DTDAm — DTS Destination Address Register..... | 397 |
| 7.10.3.3 | DTTCm — DTS Transfer Count Register..... | 398 |
| 7.10.3.4 | DTTCTm — DTS Transfer Control Register..... | 399 |
| 7.10.3.5 | DTRSAm — DTS Reload Source Address Register..... | 402 |
| 7.10.3.6 | DTRDAm — DTS Reload Destination Address Register..... | 403 |
| 7.10.3.7 | DTRTCm — DTS Reload Transfer Count Register..... | 404 |
| 7.10.3.8 | DTTCCm — DTS Transfer Count Compare Register..... | 405 |
| 7.10.3.9 | DTFSLm — DTSFSL Operation Setting Register..... | 406 |
| 7.10.3.10 | DTFSTm — DTSFSL Transfer Request Status Register..... | 407 |
| 7.10.3.11 | DTFSSm — DTSFSL Transfer Request Set Register..... | 408 |
| 7.10.3.12 | DTFSCm — DTSFSL Transfer Request Clear Register..... | 409 |
| 7.11 | DMAC/DTS Trigger Select Registers..... | 410 |
| 7.11.1 | DTS Trigger Select Register Descriptions..... | 410 |
| 7.11.1.1 | DMACTRGSEL0 — DMAC Primary/Secondary Select Register 0..... | 410 |
| 7.11.1.2 | DMACTRGSEL1 — DMAC Primary/Secondary Select Register 1..... | 412 |
| 7.11.1.3 | DTSTRGSEL0 — DTS Primary/Secondary Select Register 0..... | 414 |
| 7.11.1.4 | DTSTRGSEL1 — DTS Primary/Secondary Select Register 1..... | 416 |
| Section 8 | Reset Controller..... | 418 |
| 8.1 | Features..... | 418 |
| 8.2 | Input/Output Pins..... | 419 |
| 8.3 | Register Description..... | 420 |
| 8.3.1 | RESF — Reset Factor Register..... | 421 |
| 8.3.2 | RESFC — Reset Factor Clear Register..... | 423 |
| 8.3.3 | SWSRESA0 — Software System Reset Request Register 0..... | 424 |
| 8.3.4 | SWARESAS0 — Software Application Reset Request Register 0..... | 425 |
| 8.3.5 | RESC — Reset Configuration Register..... | 426 |
| 8.3.6 | STAC_DTSRAM — RAM Initialization Mode Control Register for DTS RAM..... | 427 |
| 8.3.7 | STAC_GRAM — RAM Initialization Mode Control Register for Global RAM..... | 428 |
| 8.3.8 | STAC_LM0 — RAM Initialization Mode Control Register for Local RAM..... | 429 |
| 8.3.9 | STAC_LM10 — RAM Initialization Mode Control Register for CSIH..... | 430 |
| 8.4 | Operation..... | 431 |
| 8.4.1 | Reset Categories..... | 431 |
| 8.4.2 | Reset Sources..... | 432 |
| 8.4.3 | Reset Flags..... | 434 |
| 8.4.4 | Reading Option Bytes from FLASH..... | 434 |
| 8.4.5 | Filed BIST..... | 434 |
| 8.4.6 | RAM initialization..... | 434 |
| 8.4.7 | Reset Mask function..... | 434 |
| 8.4.8 | Reset Output (RESETOUT)..... | 435 |
| Section 9 | Power Supply Circuit..... | 436 |
| 9.1 | Features..... | 436 |
| 9.2 | External Pin List..... | 437 |

| | | |
|--|--|------------|
| 9.3 | Block Diagram | 438 |
| 9.3.1 | Single Power Supply | 438 |
| 9.3.2 | Dual Power Supply | 439 |
| 9.4 | Connection Example..... | 440 |
| 9.4.1 | Single Power Supply | 440 |
| 9.4.2 | Dual Power Supply | 440 |
| 9.5 | Power Up/Down Timing | 441 |
| Section 10 Core Voltage Monitor | | 442 |
| 10.1 | Overview..... | 442 |
| 10.1.1 | Functional Overview | 442 |
| 10.1.2 | Block Diagram | 443 |
| 10.2 | Input/Output Pins | 444 |
| 10.3 | Registers..... | 445 |
| 10.3.1 | List of Registers..... | 445 |
| 10.3.2 | CVMF — CVM Factor Register | 446 |
| 10.3.3 | CVMFC — CVM Factor Clear Register | 447 |
| 10.3.4 | CVMDE — CVM Detection Enable Register | 448 |
| 10.3.5 | CVMDEW — CVM Detection Enable Set Register | 449 |
| 10.3.6 | CVMDMASK — CVM Detection Output Mask Register | 450 |
| 10.3.7 | CVMDIAG — CVM DIAG Mode Setting Register | 451 |
| 10.3.8 | CVMMON — CVM Monitor Register | 452 |
| 10.4 | Operation..... | 453 |
| 10.4.1 | CVM Basic Function | 453 |
| 10.4.2 | CVM Function in Field BIST operation and Serial Programming Mode (Mask of CVMOUT Pin and CVM Reset) | 456 |
| 10.4.3 | CVM Diagnosis Function | 456 |
| 10.5 | Usage Notes | 458 |
| Section 11 Temperature Sensor | | 459 |
| 11.1 | Features of RH850/P1M-E Temperature Sensor | 459 |
| 11.1.1 | Number of Units..... | 459 |
| 11.1.2 | Register Base Address..... | 459 |
| 11.1.3 | Clock Supply..... | 459 |
| 11.1.4 | Interrupt Request..... | 459 |
| 11.1.5 | Reset Sources | 459 |
| 11.1.6 | External Input/Output Signals..... | 459 |
| 11.2 | Overview..... | 460 |
| 11.2.1 | Functional Overview | 460 |
| 11.2.2 | Block Diagram | 460 |
| 11.3 | Register | 461 |
| 11.3.1 | Register List..... | 461 |
| 11.3.2 | TSN0CR — Temperature Sensor Control Register..... | 461 |
| 11.3.3 | TSN0STAT — Temperature Sensor Status Register | 462 |

| | | |
|-------------------|---|------------|
| 11.3.4 | TSN0DIAG — Temperature Sensor Diagnosis Control Register | 463 |
| 11.3.5 | TSNREFD — Temperature Sensor Reference Temperature Storage Register | 464 |
| 11.4 | Functions | 465 |
| 11.4.1 | Temperature Measurement | 465 |
| 11.4.2 | Temperature Error Notification | 466 |
| 11.4.3 | Self-Diagnosis Function..... | 466 |
| 11.5 | Calculating the Temperature | 467 |
| Section 12 | Clock Controller | 468 |
| 12.1 | Features..... | 468 |
| 12.1.1 | External Input/Output Pins..... | 468 |
| 12.2 | Overview..... | 469 |
| 12.2.1 | Type of Clocks..... | 469 |
| 12.2.2 | Block Diagram | 470 |
| 12.3 | Register Description | 471 |
| 12.3.1 | Writing Protection to Registers | 471 |
| 12.3.2 | Register Overview | 471 |
| 12.3.3 | CLKD2DIV — Clock Divider 2 Divisor Register..... | 472 |
| 12.3.4 | CLKD2STAT — Clock Divider 2 Status Register | 473 |
| 12.3.5 | CLKD3DIV — Clock Divider 3 Divisor Register..... | 474 |
| 12.3.6 | CLKD3STAT — Clock Divider 3 Status Register | 475 |
| 12.3.7 | CKSC2C — Clock Selector 2 Control Register | 476 |
| 12.3.8 | CKSC2S — Clock Selector 2 Status Register..... | 477 |
| 12.3.9 | CKSC3C — Clock Selector 3 Control Register | 478 |
| 12.3.10 | CKSC3S — Clock Selector 3 Status Register..... | 479 |
| 12.3.11 | CKSC8C — Clock Selector 8 Control Register | 480 |
| 12.3.12 | CKSC8S — Clock Selector 8 Status Register..... | 481 |
| 12.4 | Operation..... | 482 |
| 12.4.1 | External Clock Output..... | 482 |
| 12.5 | Usage Notes | 483 |
| 12.5.1 | How to Connect a Crystal Oscillator..... | 483 |
| Section 13 | Clocked Serial Interface G (CSIG)..... | 484 |
| 13.1 | Features of RH850/P1M-E CSIG..... | 484 |
| 13.1.1 | Number of Units..... | 484 |
| 13.1.2 | Register Base Address..... | 484 |
| 13.1.3 | Clock Supply..... | 485 |
| 13.1.4 | Interrupt Requests | 485 |
| 13.1.5 | Reset Sources | 485 |
| 13.1.6 | External Input/Output Signals..... | 486 |
| 13.1.7 | Data Consistency Check | 486 |
| 13.1.8 | Combinations of Pins and Ports | 486 |
| 13.2 | Overview..... | 487 |
| 13.2.1 | Functional Overview | 487 |

| | | |
|-----------|---|-----|
| 13.2.2 | Functional Description | 488 |
| 13.2.3 | Block Diagram | 489 |
| 13.3 | Registers..... | 490 |
| 13.3.1 | List of Registers..... | 490 |
| 13.3.2 | CSIGNCTL0 — CSIGN Control Register 0..... | 491 |
| 13.3.3 | CSIGNCTL1 — CSIGN Control Register 1..... | 492 |
| 13.3.4 | CSIGNCTL2 — CSIGN Control Register 2..... | 494 |
| 13.3.5 | CSIGNSTR0 — CSIGN Status Register 0..... | 495 |
| 13.3.6 | CSIGNSTCR0 — CSIGN Status Clear Register 0 | 497 |
| 13.3.7 | CSIGNBCTL0 — CSIGN Rx-only Mode Control Register 0 | 498 |
| 13.3.8 | CSIGNCFG0 — CSIGN Configuration Register 0 | 499 |
| 13.3.9 | CSIGNTX0W — CSIGN Transmission Register 0 for Word Access | 501 |
| 13.3.10 | CSIGNTX0H — CSIGN Transmission Register 0 for Half Word Access..... | 502 |
| 13.3.11 | CSIGNRX0 — CSIGN Reception Register 0..... | 502 |
| 13.3.12 | List of Caution..... | 503 |
| 13.4 | Interrupt Sources | 504 |
| 13.4.1 | Interrupt Delay | 504 |
| 13.4.2 | INTCSIG0IC (Communication Status Interrupt) | 505 |
| 13.4.3 | INTCSIG0IR (Reception Status Interrupt) | 506 |
| 13.4.4 | INTCSIG0IRE (Communication Error Interrupt) | 506 |
| 13.5 | Operation | 507 |
| 13.5.1 | Master/Slave Mode..... | 507 |
| 13.5.1.1 | Master Mode | 507 |
| 13.5.1.2 | Slave Mode | 508 |
| 13.5.2 | Master/Slave Connections..... | 509 |
| 13.5.2.1 | One Master and One Slave..... | 509 |
| 13.5.3 | Transmission Clock Selection | 510 |
| 13.5.4 | Data Transfer Modes..... | 511 |
| 13.5.4.1 | Transmit-Only Mode..... | 511 |
| 13.5.4.2 | Receive-Only Mode..... | 511 |
| 13.5.4.3 | Transmit/Receive Mode | 511 |
| 13.5.5 | Data Length Selection | 512 |
| 13.5.5.1 | Data Length Selection Without Extended Length..... | 512 |
| 13.5.5.2 | Data Length Selection with Extended Data Length..... | 512 |
| 13.5.6 | Serial Data Direction Selection Function | 514 |
| 13.5.7 | Communication Timing in Slave Mode | 515 |
| 13.5.8 | Handshake Function..... | 516 |
| 13.5.8.1 | Slave Mode | 516 |
| 13.5.8.2 | Master Mode | 517 |
| 13.5.9 | Loop-Back Mode | 518 |
| 13.5.10 | Error Detection | 519 |
| 13.5.10.1 | Data Consistency Check..... | 519 |
| 13.5.10.2 | Parity Check | 520 |
| 13.5.10.3 | Overrun Error | 521 |
| 13.6 | Operating Procedures..... | 523 |
| 13.6.1 | Transmission/Reception by DMA in Master Mode | 523 |

| | | |
|------------|---|-----|
| Section 14 | Clocked Serial Interface H (CSIH) | 525 |
| 14.1 | Features of RH850/P1M-E CSIH | 525 |
| 14.1.1 | Number of Units | 525 |
| 14.1.2 | Register Base Address | 526 |
| 14.1.3 | Clock Supply | 526 |
| 14.1.4 | Interrupt Requests | 527 |
| 14.1.5 | Reset Sources | 527 |
| 14.1.6 | External Input/Output Signals | 528 |
| 14.1.7 | Data Consistency Check | 529 |
| 14.1.8 | Combinations of Pins and Ports | 530 |
| 14.2 | Overview | 532 |
| 14.2.1 | Functional Overview | 532 |
| 14.2.2 | Functional Overview Description | 533 |
| 14.2.3 | Block Diagram | 534 |
| 14.3 | Registers | 535 |
| 14.3.1 | List of Registers | 535 |
| 14.3.2 | CSIHnCTL0 — CSIHn Control Register 0 | 536 |
| 14.3.3 | CSIHnCTL1 — CSIHn Control Register 1 | 537 |
| 14.3.4 | CSIHnCTL2 — CSIHn Control Register 2 | 540 |
| 14.3.5 | CSIHnSTR0 — CSIHn Status Register 0 | 542 |
| 14.3.6 | CSIHnSTCR0 — CSIHn Status Clear Register 0 | 546 |
| 14.3.7 | CSIHnMCTL0 — CSIHn Memory Control Register 0 | 547 |
| 14.3.8 | CSIHnMCTL1 — CSIHn Memory Control Register 1 | 548 |
| 14.3.9 | CSIHnMCTL2 — CSIHn Memory Control Register 2 | 549 |
| 14.3.10 | CSIHnMRWP0 — CSIHn Memory Read/Write Pointer Register 0 | 551 |
| 14.3.11 | CSIHnCFGx — CSIHn Configuration Register x | 553 |
| 14.3.12 | CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access | 558 |
| 14.3.13 | CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access | 560 |
| 14.3.14 | CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access | 561 |
| 14.3.15 | CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access | 562 |
| 14.3.16 | CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3) | 563 |
| 14.3.17 | SELCSIHDMA — CSIH DMA Select Register | 564 |
| 14.3.18 | List of Cautions | 566 |
| 14.4 | Interrupt Sources | 569 |
| 14.4.1 | Overview | 569 |
| 14.4.2 | Interrupt Delay | 570 |
| 14.4.3 | INTCSIHnIC (Communication Status Interrupt) | 571 |
| 14.4.3.1 | INTCSIHnIC in Direct Access Mode | 572 |
| 14.4.3.2 | INTCSIHnIC in FIFO Mode | 573 |
| 14.4.3.3 | INTCSIHnIC in Job Mode | 574 |
| 14.4.4 | INTCSIHnIR (Receive Status Interrupt) | 575 |
| 14.4.4.1 | INTCSIHnIR in Direct Access Mode | 575 |
| 14.4.4.2 | INTCSIHnIR in Dual Buffer Mode | 576 |
| 14.4.5 | INTCSIHnIRE (Communication Error Interrupt) | 577 |
| 14.4.6 | INTCSIHnIJC (Job Completion Interrupt) | 578 |

| | | |
|-----------|--|-----|
| 14.5 | Operation | 579 |
| 14.5.1 | Operating Modes (Master/Slave)..... | 579 |
| 14.5.1.1 | Master mode | 579 |
| 14.5.1.2 | Slave Mode | 580 |
| 14.5.2 | Master/Slave Connections..... | 581 |
| 14.5.2.1 | One Master and One Slave..... | 581 |
| 14.5.2.2 | One Master and Multiple Slaves | 581 |
| 14.5.3 | Chip Selection (CS) Features..... | 583 |
| 14.5.3.1 | Configuration Registers..... | 583 |
| 14.5.3.2 | CS Example | 585 |
| 14.5.3.3 | Job Concept | 585 |
| 14.5.4 | Chip Select Timing Details | 586 |
| 14.5.4.1 | Changing the Clock Phase..... | 586 |
| 14.5.4.2 | Changing the Data Phase | 588 |
| 14.5.5 | Transmission Clock Selection | 589 |
| 14.5.6 | CSIH Buffer Memory | 591 |
| 14.5.6.1 | FIFO Mode | 591 |
| 14.5.6.2 | Dual Buffer Mode | 592 |
| 14.5.6.3 | Transmit-Only Buffer Mode | 592 |
| 14.5.6.4 | Direct Access Mode | 592 |
| 14.5.7 | Data Transfer Modes..... | 593 |
| 14.5.7.1 | Transmit-Only Mode..... | 593 |
| 14.5.7.2 | Receive-Only Mode..... | 593 |
| 14.5.7.3 | Transmit/Receive Mode | 593 |
| 14.5.7.4 | Summary | 593 |
| 14.5.8 | Data Length Selection | 594 |
| 14.5.8.1 | Data Length Between 2 and 16 bits | 594 |
| 14.5.8.2 | Data Length Greater than 16 Bits..... | 595 |
| 14.5.9 | Serial Data Direction Selection..... | 597 |
| 14.5.10 | Slave Select (SS) Function..... | 598 |
| 14.5.10.1 | Communication Timing Using SS Function..... | 598 |
| 14.5.10.2 | CSIH TSSO Operation | 599 |
| 14.5.11 | Handshake Function..... | 600 |
| 14.5.11.1 | Slave Mode | 600 |
| 14.5.11.2 | Master Mode | 603 |
| 14.5.12 | Error Detection | 604 |
| 14.5.12.1 | Data Consistency Check..... | 604 |
| 14.5.12.2 | Parity Check | 606 |
| 14.5.12.3 | Time-Out Error | 607 |
| 14.5.12.4 | Overflow Error | 608 |
| 14.5.12.5 | Overrun Error | 610 |
| 14.5.13 | Loop-Back Mode | 613 |
| 14.5.14 | CPU-Controlled High Priority Communication Function | 615 |
| 14.5.15 | Enforced Chip Select Idle Setting..... | 618 |
| 14.6 | Operating Procedures..... | 619 |
| 14.6.1 | Procedures in Direct Access Mode | 619 |
| 14.6.1.1 | Transmit/Receive in Master Mode when Job Mode is Disabled..... | 619 |
| 14.6.1.2 | Transmit/Receive in Master Mode when Job Mode is Enabled | 621 |

| | | |
|-------------------|--|------------|
| 14.6.2 | Procedures in Transmit-Only Buffer Mode | 623 |
| 14.6.2.1 | Transmit/Receive in Master Mode when Job Mode is Disabled..... | 623 |
| 14.6.2.2 | Transmit/Receive in Master Mode when Job Mode is Enabled | 625 |
| 14.6.3 | Procedures in Dual Buffer Mode | 627 |
| 14.6.3.1 | Transmit/Receive in Master Mode when Job Mode is Disabled..... | 627 |
| 14.6.3.2 | Transmit/Receive in Master Mode when Job Mode is Enabled | 629 |
| 14.6.3.3 | Transmit/Receive in Slave Mode when Job Mode is Disabled..... | 631 |
| 14.6.4 | Procedures in FIFO Mode | 633 |
| 14.6.4.1 | Transmit/Receive in Master Mode when Job Mode is Disabled..... | 633 |
| 14.6.4.2 | Transmit/Receive in Mater Mode when Job Mode is Enabled | 635 |
| Section 15 | Serial Communication Interface 3 (SCI3) | 637 |
| 15.1 | Features of RH850/P1M-E SCI3 | 637 |
| 15.1.1 | Number of Units and Channels | 637 |
| 15.1.2 | Register Base Address..... | 638 |
| 15.1.3 | Clock Supply..... | 638 |
| 15.1.4 | Interrupt Request..... | 638 |
| 15.1.5 | Reset Sources | 639 |
| 15.1.6 | External Input/Output Signals..... | 639 |
| 15.1.7 | Combination of Pins and Ports | 640 |
| 15.2 | Outline of Functions..... | 641 |
| 15.2.1 | Serial Communication Modes..... | 641 |
| 15.2.2 | Block Diagram | 642 |
| 15.3 | Register Descriptions..... | 643 |
| 15.3.1 | SCI3nRSR — Receive Shift Register..... | 644 |
| 15.3.2 | SCI3nRDR — Receive Data Register | 644 |
| 15.3.3 | SCI3nTDR — Transmit Data Register..... | 644 |
| 15.3.4 | SCI3nTSR — Transmit Shift Register | 644 |
| 15.3.5 | SCI3nSMR — Serial Mode Register | 645 |
| 15.3.6 | SCI3nSCR — Serial Control Register | 646 |
| 15.3.7 | SCI3nSSR — Serial Status Register..... | 648 |
| 15.3.8 | SCI3nSCMR — Serial Transfer Format Register | 650 |
| 15.3.9 | SCI3nSEMR — Serial Extended Mode Register..... | 651 |
| 15.3.10 | SCI3nBRR — Bit Rate Register | 652 |
| 15.3.11 | SCI3nMDDR — Modulation Duty Register..... | 655 |
| 15.4 | Operation in Asynchronous Mode | 656 |
| 15.4.1 | Transmission/Reception Format..... | 657 |
| 15.4.2 | Receive Data Sampling Timing and Reception Margin | 658 |
| 15.4.3 | Clock..... | 659 |
| 15.4.4 | Double-Speed Operation..... | 659 |
| 15.4.5 | SCI3 Initialization (Asynchronous Mode)..... | 660 |
| 15.4.6 | Serial Data Transmission (Asynchronous Mode) | 661 |
| 15.4.7 | Serial Data Reception (Asynchronous Mode) | 664 |
| 15.5 | Multi-Processor Communication Function | 668 |
| 15.5.1 | Overview and Sample Connection | 668 |

| | | |
|-------------------|--|------------|
| 15.5.2 | Multi-Processor Serial Data Transmission | 669 |
| 15.5.3 | Multi-Processor Serial Data Reception..... | 670 |
| 15.6 | Operation in Clock Synchronous Mode | 674 |
| 15.6.1 | Clock..... | 674 |
| 15.6.2 | SCI3 Initialization (Clock Synchronous Mode) | 675 |
| 15.6.3 | Serial Data Transmission (Clock Synchronous Mode)..... | 676 |
| 15.6.4 | Serial Data Reception (Clock Synchronous Mode) | 679 |
| 15.6.5 | Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode) | 681 |
| 15.7 | Bit Rate Modulation Function..... | 683 |
| 15.8 | Interrupt Sources | 684 |
| 15.9 | Usage Notes | 685 |
| 15.9.1 | Break Detection and Processing | 685 |
| 15.9.2 | Mark State and Break Output | 685 |
| 15.9.3 | Receive Error Flags and Transmit Operations in Clock Synchronous Mode | 685 |
| 15.9.4 | Relationship between Writing to SCI3nTDR and the TDRE Flag..... | 685 |
| 15.9.5 | Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode | 686 |
| 15.9.6 | External Clock Input in Clock Synchronous Mode..... | 686 |
| Section 16 | LIN/UART Interface (RLIN3)..... | 687 |
| 16.1 | Features of RH850/P1M-E RLIN3 | 687 |
| 16.1.1 | Number of Units and Channels | 687 |
| 16.1.2 | Register Base Address..... | 688 |
| 16.1.3 | Clock Supply..... | 688 |
| 16.1.4 | Interrupt Request..... | 688 |
| 16.1.5 | Reset Sources | 688 |
| 16.1.6 | External Input/output Signals..... | 689 |
| 16.1.7 | Combination of Pin Name and Port Name | 689 |
| 16.2 | Overview..... | 690 |
| 16.2.1 | Functional Overview | 690 |
| 16.2.2 | Block Diagram | 693 |
| 16.2.3 | Description of Blocks | 693 |
| 16.3 | Registers..... | 694 |
| 16.3.1 | List of Registers..... | 694 |
| 16.3.2 | LIN Master Related Registers | 695 |
| 16.3.2.1 | RLN3nLWBR — LIN Wake-Up Baud Rate Select Register | 695 |
| 16.3.2.2 | RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register | 696 |
| 16.3.2.3 | RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register | 697 |
| 16.3.2.4 | RLN3nLSTC — LIN Self-Test Control Register | 698 |
| 16.3.2.5 | RLN3nLMD — LIN Mode Register | 699 |
| 16.3.2.6 | RLN3nLBFC — LIN Break Field Configuration Register..... | 701 |
| 16.3.2.7 | RLN3nLSC — LIN Space Configuration Register | 702 |
| 16.3.2.8 | RLN3nLWUP — LIN Wake-Up Configuration Register..... | 703 |
| 16.3.2.9 | RLN3nLIE — LIN Interrupt Enable Register..... | 704 |
| 16.3.2.10 | RLN3nLEDE — LIN Error Detection Enable Register..... | 706 |
| 16.3.2.11 | RLN3nLCUC — LIN Control Register | 708 |
| 16.3.2.12 | RLN3nLTRC — LIN Transmission Control Register | 709 |

| | | |
|-----------|---|-----|
| 16.3.2.13 | RLN3nLMST — LIN Mode Status Register | 710 |
| 16.3.2.14 | RLN3nLST — LIN Status Register | 711 |
| 16.3.2.15 | RLN3nLEST — LIN Error Status Register | 713 |
| 16.3.2.16 | RLN3nLDFC — LIN Data Field Configuration Register | 715 |
| 16.3.2.17 | RLN3nLIDB — LIN ID Buffer Register | 717 |
| 16.3.2.18 | RLN3nLCBR — LIN Checksum Buffer Register | 718 |
| 16.3.2.19 | RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8) | 719 |
| 16.3.3 | UART Related Registers | 721 |
| 16.3.3.1 | RLN3nLWBR — LIN Wake-Up Baud Rate Select Register | 721 |
| 16.3.3.2 | RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register | 722 |
| 16.3.3.3 | RLN3nLMD — UART Mode Register | 723 |
| 16.3.3.4 | RLN3nLBFC — UART Configuration Register | 724 |
| 16.3.3.5 | RLN3nLSC — UART Space Configuration Register | 726 |
| 16.3.3.6 | RLN3nLEDE — UART Error Detection Enable Register | 727 |
| 16.3.3.7 | RLN3nLCUC — UART Control Register | 728 |
| 16.3.3.8 | RLN3nLTRC — UART Transmission Control Register | 729 |
| 16.3.3.9 | RLN3nLMST — UART Mode Status Register | 730 |
| 16.3.3.10 | RLN3nLST — UART Status Register | 731 |
| 16.3.3.11 | RLN3nLEST — UART Error Status Register | 733 |
| 16.3.3.12 | RLN3nLDFC — UART Data Field Configuration Register | 735 |
| 16.3.3.13 | RLN3nLIDB — UART ID Buffer Register | 736 |
| 16.3.3.14 | RLN3nLUDB0 — UART Data Buffer 0 Register | 736 |
| 16.3.3.15 | RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8) | 737 |
| 16.3.3.16 | RLN3nLUOER — UART Operation Enable Register | 738 |
| 16.3.3.17 | RLN3nLUOR1 — UART Option Register 1 | 739 |
| 16.3.3.18 | RLN3nLUTDR — UART Transmission Data Register | 741 |
| 16.3.3.19 | RLN3nLURDR — UART Reception Data Register | 742 |
| 16.3.3.20 | RLN3nLUWTD — UART Wait Transmission Data Register | 743 |
| 16.4 | Interrupt Sources | 744 |
| 16.5 | Modes | 745 |
| 16.6 | LIN Reset Mode | 747 |
| 16.7 | LIN Mode | 748 |
| 16.7.1 | LIN Master Mode | 750 |
| 16.7.1.1 | Header Transmission | 750 |
| 16.7.1.2 | Response Transmission | 751 |
| 16.7.1.3 | Response Reception | 752 |
| 16.7.2 | Data Transmission/Reception | 753 |
| 16.7.2.1 | Data Transmission | 753 |
| 16.7.2.2 | Data Reception | 754 |
| 16.7.3 | Transmission/Reception Data Buffering | 755 |
| 16.7.3.1 | Transmission of LIN Frames | 755 |
| 16.7.3.2 | Reception of LIN Frames | 756 |
| 16.7.3.3 | Multi-Byte Response Transmission/Reception Function | 757 |
| 16.7.4 | Wake-up Transmission/Reception | 758 |
| 16.7.4.1 | Wake-up Transmission | 758 |
| 16.7.4.2 | Wake-up Reception | 759 |
| 16.7.4.3 | Wakeup Collision | 759 |
| 16.7.5 | Status | 760 |

| | | |
|-------------------|---|------------|
| 16.7.6 | Error Status | 761 |
| 16.7.6.1 | LIN Master Mode..... | 761 |
| 16.8 | UART Mode..... | 763 |
| 16.8.1 | Transmission | 763 |
| 16.8.1.1 | Continuous Transmission..... | 765 |
| 16.8.1.2 | UART Buffer Transmission..... | 766 |
| 16.8.1.3 | Data Transmission | 768 |
| 16.8.1.4 | Transmission Start Wait Function | 769 |
| 16.8.2 | Reception | 770 |
| 16.8.2.1 | Data Reception..... | 771 |
| 16.8.3 | Expansion Bits..... | 772 |
| 16.8.3.1 | Expansion Bit Transmission | 772 |
| 16.8.3.2 | Expansion Bit Reception | 772 |
| 16.8.3.3 | Expansion Bit Reception (with Expansion Bit Comparison) | 773 |
| 16.8.3.4 | Expansion Bit Reception (with Data Comparison) | 774 |
| 16.8.4 | Status | 775 |
| 16.8.5 | Error Status | 776 |
| 16.9 | LIN Self-Test Mode..... | 777 |
| 16.9.1 | Transitioning to LIN Self-Test Mode..... | 779 |
| 16.9.2 | Transmission in LIN Master Self-Test Mode | 780 |
| 16.9.3 | Reception in LIN Master Self-Test Mode..... | 781 |
| 16.9.4 | Terminating LIN Self-Test Mode..... | 782 |
| 16.10 | Baud Rate Generator..... | 783 |
| 16.10.1 | LIN Master Mode..... | 783 |
| 16.10.2 | UART Mode..... | 785 |
| 16.11 | Noise Filter..... | 786 |
| Section 17 | CANFD Interface (RS-CANFD)..... | 788 |
| 17.1 | Features of RH850/P1M-E RS-CANFD..... | 788 |
| 17.1.1 | Number of Units and Channels | 788 |
| 17.1.2 | Register Base Address..... | 791 |
| 17.1.3 | Clock Supply..... | 791 |
| 17.1.4 | Interrupt Requests | 792 |
| 17.1.5 | Reset Source..... | 793 |
| 17.1.6 | External Input/Output Signals..... | 793 |
| 17.1.7 | Combinations of Pins and Ports | 793 |
| 17.2 | Overview..... | 794 |
| 17.2.1 | Functional Overview | 794 |
| 17.2.2 | Interface Modes..... | 796 |
| 17.2.3 | Block Diagram | 797 |
| 17.3 | Registers (Classical CAN Mode) | 798 |
| 17.3.1 | List of Registers..... | 798 |
| 17.3.2 | Details of Interface Mode-Related Registers..... | 802 |
| 17.3.2.1 | RSCANnGRMCFG — Global Interface Mode Select Register | 802 |
| 17.3.3 | Details of Channel-Related Registers | 803 |

| | | |
|----------|---|-----|
| 17.3.3.1 | RSCANnCMCFG — Channel m Configuration Register (m = 0 to 2) | 803 |
| 17.3.3.2 | RSCANnCMCTR — Channel m Control Register (m = 0 to 2) | 805 |
| 17.3.3.3 | RSCANnCMSTS — Channel m Status Register (m = 0 to 2) | 810 |
| 17.3.3.4 | RSCANnCMERFL — Channel m Error Flag Register (m = 0 to 2) | 812 |
| 17.3.4 | Details of Global-Related Registers | 816 |
| 17.3.4.1 | RSCANnGCFG — Global Configuration Register | 816 |
| 17.3.4.2 | RSCANnGCTR — Global Control Register | 819 |
| 17.3.4.3 | RSCANnGSTS — Global Status Register | 821 |
| 17.3.4.4 | RSCANnGERFL — Global Error Flag Register | 823 |
| 17.3.4.5 | RSCANnGTSC — Global Timestamp Counter Register | 825 |
| 17.3.4.6 | RSCANnGTINTSTS0 — Global TX Interrupt Status Register 0 | 826 |
| 17.3.4.7 | RSCANnGFDCFG — Global FD Configuration Register | 829 |
| 17.3.5 | Details of Receive Rule-related Registers | 830 |
| 17.3.5.1 | RSCANnGAFLECTR — Receive Rule Entry Control Register | 830 |
| 17.3.5.2 | RSCANnGAFLCFG0 — Receive Rule Configuration Register 0 | 831 |
| 17.3.5.3 | RSCANnGAFLIDj — Receive Rule ID Register j (j = 0 to 15) | 832 |
| 17.3.5.4 | RSCANnGAFLMj — Receive Rule Mask Register j (j = 0 to 15) | 834 |
| 17.3.5.5 | RSCANnGAFLP0_j — Receive Rule Pointer 0 Register j (j = 0 to 15) | 835 |
| 17.3.5.6 | RSCANnGAFLP1_j — Receive Rule Pointer 1 Register j (j = 0 to 15) | 837 |
| 17.3.6 | Details of Receive Buffer-Related Registers | 838 |
| 17.3.6.1 | RSCANnRMNB — Receive Buffer Number Register | 838 |
| 17.3.6.2 | RSCANnRMNDy — Receive Buffer New Data Register y (y = 0, 1) | 839 |
| 17.3.6.3 | RSCANnRMIDq — Receive Buffer ID Register q (q = 0 to 47) | 840 |
| 17.3.6.4 | RSCANnRMPTRq — Receive Buffer Pointer Register q (q = 0 to 47) | 841 |
| 17.3.6.5 | RSCANnRMDf0_q — Receive Buffer Data Field 0 Register q (q = 0 to 47) | 842 |
| 17.3.6.6 | RSCANnRMDf1_q — Receive Buffer Data Field 1 Register q (q = 0 to 47) | 843 |
| 17.3.7 | Details of Receive FIFO Buffer-Related Registers | 844 |
| 17.3.7.1 | RSCANnRFCCx — Receive FIFO Buffer Configuration and Control Register x (x = 0 to 7) | 844 |
| 17.3.7.2 | RSCANnRFSTsx — Receive FIFO Buffer Status Register x (x = 0 to 7) | 846 |
| 17.3.7.3 | RSCANnRFPCTRx — Receive FIFO Buffer Pointer Control Register x (x = 0 to 7) | 848 |
| 17.3.7.4 | RSCANnRFIDx — Receive FIFO Buffer Access ID Register x (x = 0 to 7) | 849 |
| 17.3.7.5 | RSCANnRFPTRx — Receive FIFO Buffer Access Pointer Register x (x = 0 to 7) | 850 |
| 17.3.7.6 | RSCANnRFDF0_x — Receive FIFO Buffer Access Data Field 0 Register x (x = 0 to 7) | 851 |
| 17.3.7.7 | RSCANnRFDF1_x — Receive FIFO Buffer Access Data Field 1 Register x (x = 0 to 7) | 852 |
| 17.3.8 | Details of Transmit/Receive FIFO Buffer-Related Registers | 853 |
| 17.3.8.1 | RSCANnCFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 8) | 853 |
| 17.3.8.2 | RSCANnCFSTSk — Transmit/receive FIFO Buffer Status Register k (k = 0 to 8) | 857 |
| 17.3.8.3 | RSCANnCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register k (k = 0 to 8) | 861 |
| 17.3.8.4 | RSCANnCFIDk — Transmit/receive FIFO Buffer Access ID Register k (k = 0 to 8) | 863 |
| 17.3.8.5 | RSCANnCFPTRk — Transmit/receive FIFO Buffer Access Pointer Register k (k = 0 to 8) | 865 |
| 17.3.8.6 | RSCANnCFDF0_k — Transmit/receive FIFO Buffer Access Data Field 0 Register k (k = 0 to 8) | 867 |
| 17.3.8.7 | RSCANnCFDF1_k — Transmit/receive FIFO Buffer Access Data Field 1 | |

| | | |
|-----------|---|-----|
| | Register k (k = 0 to 8)..... | 868 |
| 17.3.9 | Details of FIFO Status-Related Registers | 869 |
| 17.3.9.1 | RSCANnFESTS — FIFO Empty Status Register..... | 869 |
| 17.3.9.2 | RSCANnFFSTS — FIFO Full Status Register | 871 |
| 17.3.9.3 | RSCANnFMSTS — FIFO Message Lost Status Register..... | 873 |
| 17.3.9.4 | RSCANnRFISTS — Receive FIFO Buffer Interrupt Flag Status Register..... | 875 |
| 17.3.9.5 | RSCANnCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register..... | 876 |
| 17.3.9.6 | RSCANnCTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register..... | 877 |
| 17.3.10 | Details of Transmit Buffer-Related Registers | 878 |
| 17.3.10.1 | RSCANnTMCp — Transmit Buffer Control Register p (p = 0 to 47) | 878 |
| 17.3.10.2 | RSCANnTMSTSp — Transmit Buffer Status Register p (p = 0 to 47) | 880 |
| 17.3.10.3 | RSCANnTMIDp — Transmit Buffer ID Register p (p = 0 to 47) | 882 |
| 17.3.10.4 | RSCANnTMPTRp — Transmit Buffer Pointer Register p (p = 0 to 47) | 884 |
| 17.3.10.5 | RSCANnTMDF0_p — Transmit Buffer Data Field 0 Register p (p = 0 to 47) | 886 |
| 17.3.10.6 | RSCANnTMDF1_p — Transmit Buffer Data Field 1 Register p (p = 0 to 47) | 887 |
| 17.3.10.7 | RSCANnTMIECy — Transmit Buffer Interrupt Enable Configuration Register y (y = 0, 1)..... | 888 |
| 17.3.11 | Details of Transmit Buffer Status-Related Registers | 890 |
| 17.3.11.1 | RSCANnTMTRSTSy — Transmit Buffer Transmit Request Status Register y (y = 0, 1)..... | 890 |
| 17.3.11.2 | RSCANnTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y (y = 0, 1)..... | 892 |
| 17.3.11.3 | RSCANnTMTCASTSy — Transmit Buffer Transmit Complete Status Register y (y = 0, 1)..... | 894 |
| 17.3.11.4 | RSCANnTMTASTSy — Transmit Buffer Transmit Abort Status Register y (y = 0, 1)..... | 896 |
| 17.3.12 | Details of Transmit Queue-Related Registers | 898 |
| 17.3.12.1 | RSCANnTXQCCm — Transmit Queue Configuration and Control Register m (m = 0 to 2)..... | 898 |
| 17.3.12.2 | RSCANnTXQSTSm — Transmit Queue Status Register m (m = 0 to 2)..... | 900 |
| 17.3.12.3 | RSCANnTXQPCTRM — Transmit Queue Pointer Control Register m (m = 0 to 2)..... | 902 |
| 17.3.13 | Details of Transmit history-related Registers..... | 903 |
| 17.3.13.1 | RSCANnTHLCCm — Transmit History Configuration and Control Register m (m = 0 to 2)..... | 903 |
| 17.3.13.2 | RSCANnTHLSTSm — Transmit History Status Register m (m = 0 to 2)..... | 905 |
| 17.3.13.3 | RSCANnTHLPCTRM — Transmit History Pointer Control Register m (m = 0 to 2)..... | 907 |
| 17.3.13.4 | RSCANnTHLACCM — Transmit History Access Register m (m = 0 to 2)..... | 908 |
| 17.3.14 | Details of Test-Related Registers..... | 910 |
| 17.3.14.1 | RSCANnGTSTCFG — Global Test Configuration Register..... | 910 |
| 17.3.14.2 | RSCANnGTSTCTR — Global Test Control Register..... | 912 |
| 17.3.14.3 | RSCANnGLOCKK — Global Lock Key Register..... | 913 |
| 17.3.14.4 | RSCANnRPGACCr — RAM Test Page Access Register r (r = 0 to 63) | 914 |
| 17.3.15 | Details of Mode Read Register..... | 915 |
| 17.3.15.1 | RSCANnCANFDMDR — CAN FD Mode Read Register | 915 |
| 17.4 | Registers (CAN FD Mode)..... | 916 |
| 17.4.1 | List of Registers..... | 916 |

| | | |
|----------|--|-----|
| 17.4.2 | Details of Interface Mode-Related Registers..... | 920 |
| 17.4.2.1 | RSCFDnCFDGRMCFG — Global Interface Mode Select Register | 920 |
| 17.4.3 | Details of Channel-Related Registers | 921 |
| 17.4.3.1 | RSCFDnCFDCmNCFG — Channel m Nominal Bit Rate Configuration Register (m = 0 to 2)..... | 921 |
| 17.4.3.2 | RSCFDnCFDCmCTR — Channel m Control Register (m = 0 to 2) | 923 |
| 17.4.3.3 | RSCFDnCFDCmSTS — Channel m Status Register (m = 0 to 2)..... | 928 |
| 17.4.3.4 | RSCFDnCFDCmERFL — Channel m Error Flag Register (m = 0 to 2)..... | 931 |
| 17.4.3.5 | RSCFDnCFDCmDCFG — Channel m Data Bit Rate Configuration Register (m = 0 to 2)..... | 935 |
| 17.4.3.6 | RSCFDnCFDCmFDCFG — Channel m CAN FD Configuration Register (m = 0 to 2)..... | 938 |
| 17.4.3.7 | RSCFDnCFDCmFDCTR — Channel m CAN FD Control Register (m = 0 to 2) ... | 942 |
| 17.4.3.8 | RSCFDnCFDCmFDSTS — Channel m CAN FD Status Register (m = 0 to 2)..... | 943 |
| 17.4.3.9 | RSCFDnCFDCmFDCRC — Channel m CAN FD CRC Register (m = 0 to 2) | 946 |
| 17.4.4 | Details of Global-Related Registers | 948 |
| 17.4.4.1 | RSCFDnCFDGCFG — Global Configuration Register | 948 |
| 17.4.4.2 | RSCFDnCFDGCTR — Global Control Register | 952 |
| 17.4.4.3 | RSCFDnCFDGSTS — Global Status Register | 954 |
| 17.4.4.4 | RSCFDnCFDGERFL — Global Error Flag Register | 956 |
| 17.4.4.5 | RSCFDnCFDGTSC — Global Timestamp Counter Register..... | 958 |
| 17.4.4.6 | RSCFDnCFDGTINTSTS0 — Global TX Interrupt Status Register 0 | 959 |
| 17.4.4.7 | RSCFDnCFDGFDCFG — Global FD Configuration Register..... | 962 |
| 17.4.5 | Details of Receive Rule-related Registers..... | 963 |
| 17.4.5.1 | RSCFDnCFDGAFLECTR — Receive Rule Entry Control Register | 963 |
| 17.4.5.2 | RSCFDnCFDGAFLCFG0 — Receive Rule Configuration Register 0..... | 964 |
| 17.4.5.3 | RSCFDnCFDGAFLIDj — Receive Rule ID Register j (j = 0 to 15)..... | 966 |
| 17.4.5.4 | RSCFDnCFDGAFLMj — Receive Rule Mask Register j (j = 0 to 15) | 968 |
| 17.4.5.5 | RSCFDnCFDGAFLP0_j — Receive Rule Pointer 0 Register j (j = 0 to 15) | 969 |
| 17.4.5.6 | RSCFDnCFDGAFLP1_j — Receive Rule Pointer 1 Register j (j = 0 to 15) | 971 |
| 17.4.6 | Details of Receive Buffer-related Registers..... | 972 |
| 17.4.6.1 | RSCFDnCFDRMNB — Receive Buffer Number Register..... | 972 |
| 17.4.6.2 | RSCFDnCFDRMNDy — Receive Buffer New Data Register y (y = 0, 1) | 973 |
| 17.4.6.3 | RSCFDnCFDRMIDq — Receive Buffer ID Register q (q = 0 to 47)..... | 974 |
| 17.4.6.4 | RSCFDnCFDRMPTRq — Receive Buffer Pointer Register q (q = 0 to 47) | 975 |
| 17.4.6.5 | RSCFDnCFDRMFDSTSq — Receive Buffer CAN FD Status Register q (q = 0 to 47)..... | 977 |
| 17.4.6.6 | RSCFDnCFDRMDFb_q — Receive Buffer Data Field b Register q (b = 0 to 4, q = 0 to 47)..... | 978 |
| 17.4.7 | Details of Receive FIFO Buffer-related Registers | 979 |
| 17.4.7.1 | RSCFDnCFDRFCCx — Receive FIFO Buffer Configuration and Control Register x (x = 0 to 7)..... | 979 |
| 17.4.7.2 | RSCFDnCFDRFSTSx — Receive FIFO Buffer Status Register x (x = 0 to 7)..... | 981 |
| 17.4.7.3 | RSCFDnCFDRFPCTRx — Receive FIFO Buffer Pointer Control Register x (x = 0 to 7)..... | 983 |
| 17.4.7.4 | RSCFDnCFDRFIDx — Receive FIFO Buffer Access ID Register x (x = 0 to 7) ... | 984 |
| 17.4.7.5 | RSCFDnCFDRFPTRx — Receive FIFO Buffer Access Pointer Register x (x = 0 to 7)..... | 985 |
| 17.4.7.6 | RSCFDnCFDRFFDSTSx — Receive FIFO CAN FD Status Register x (x = 0 to 7)..... | 987 |
| 17.4.7.7 | RSCFDnCFDRFDFd_x — Receive FIFO Buffer Access Data Field d Register x | |

| | | |
|-----------|---|------|
| | (d = 0 to 15, x = 0 to 7) | 988 |
| 17.4.8 | Transmit/Receive FIFO Buffer Related Registers | 989 |
| 17.4.8.1 | RSCFDnCFDCFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 8) | 989 |
| 17.4.8.2 | RSCFDnCFDCFSTSk — Transmit/receive FIFO Buffer Status Register k (k = 0 to 8) | 993 |
| 17.4.8.3 | RSCFDnCFDCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register k (k = 0 to 8) | 996 |
| 17.4.8.4 | RSCFDnCFDCFIDk — Transmit/receive FIFO Buffer Access ID Register k (k = 0 to 8) | 998 |
| 17.4.8.5 | RSCFDnCFDCFPTRk — Transmit/receive FIFO Buffer Access Pointer Register k (k = 0 to 8) | 1000 |
| 17.4.8.6 | RSCFDnCFDCFFDCSTSk — Transmit/Receive FIFO CAN FD Configuration / Status Register k (k = 0 to 8) | 1002 |
| 17.4.8.7 | RSCFDnCFDCFDf_d_k — Transmit/receive FIFO Buffer Access Data Field d Register k (d = 0 to 15, k = 0 to 8) | 1004 |
| 17.4.9 | Details of FIFO Status-related Registers | 1005 |
| 17.4.9.1 | RSCFDnCFDFESTS — FIFO Empty Status Register | 1005 |
| 17.4.9.2 | RSCFDnCFDFFSTS — FIFO Full Status Register | 1007 |
| 17.4.9.3 | RSCFDnCFDFMSTS — FIFO Message Lost Status Register | 1009 |
| 17.4.9.4 | RSCFDnCFDRFISTS — Receive FIFO Buffer Interrupt Flag Status Register ... | 1011 |
| 17.4.9.5 | RSCFDnCFDCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register | 1012 |
| 17.4.9.6 | RSCFDnCFDCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register | 1013 |
| 17.4.10 | Details of FIFO DMA-Related Registers | 1014 |
| 17.4.10.1 | RSCFDnCFDCDTCT — DMA Enable Register | 1014 |
| 17.4.10.2 | RSCFDnCFDCDTSTS — DMA Status Register | 1016 |
| 17.4.11 | Details of Transmit Buffer-related Registers | 1018 |
| 17.4.11.1 | RSCFDnCFDTMCp — Transmit Buffer Control Register p (p = 0 to 47) | 1018 |
| 17.4.11.2 | RSCFDnCFDTMSTSp — Transmit Buffer Status Register p (p = 0 to 47) | 1020 |
| 17.4.11.3 | RSCFDnCFDTMIDp — Transmit Buffer ID Register p (p = 0 to 47) | 1022 |
| 17.4.11.4 | RSCFDnCFDTMPTRp — Transmit Buffer Pointer Register p (p = 0 to 47) | 1024 |
| 17.4.11.5 | RSCFDnCFDTMFDCTRp — Transmit Buffer CAN FD Configuration Register p (p = 0 to 47) | 1026 |
| 17.4.11.6 | RSCFDnCFDTMDFb_p — Transmit Buffer Data Field b Register p (b = 0 to 4, p = 0 to 47) | 1028 |
| 17.4.11.7 | RSCFDnCFDTMIECy — Transmit Buffer Interrupt Enable Configuration Register y (y = 0, 1) | 1029 |
| 17.4.12 | Details of Transmit Buffer Status-related Registers | 1031 |
| 17.4.12.1 | RSCFDnCFDTMTRSTSy — Transmit Buffer Transmit Request Status Register y (y = 0, 1) | 1031 |
| 17.4.12.2 | RSCFDnCFDTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y (y = 0, 1) | 1033 |
| 17.4.12.3 | RSCFDnCFDTMTCSTSy — Transmit Buffer Transmit Complete Status Register y (y = 0, 1) | 1035 |
| 17.4.12.4 | RSCFDnCFDTMTASTSy — Transmit Buffer Transmit Abort Status Register y (y = 0, 1) | 1037 |
| 17.4.13 | Details of Transmit Queue-related Registers | 1039 |
| 17.4.13.1 | RSCFDnCFDTXQCCm — Transmit Queue Configuration and Control Register m (m = 0 to 2) | 1039 |
| 17.4.13.2 | RSCFDnCFDTXQSTSm — Transmit Queue Status Register m (m = 0 to 2) | 1041 |

| | | |
|-----------|--|------|
| 17.4.13.3 | RSCFDnCFDTXQPCTR _m — Transmit Queue Pointer Control Register _m (_m = 0 to 2) | 1043 |
| 17.4.14 | Details of Transmit History-related Registers | 1044 |
| 17.4.14.1 | RSCFDnCFDTHLCC _m — Transmit History Configuration and Control Register _m (_m = 0 to 2) | 1044 |
| 17.4.14.2 | RSCFDnCFDTHLST _{Sm} — Transmit History Status Register _m (_m = 0 to 2) | 1046 |
| 17.4.14.3 | RSCFDnCFDTHLPCTR _m — Transmit History Pointer Control Register _m (_m = 0 to 2) | 1048 |
| 17.4.14.4 | RSCFDnCFDTHLACC _m — Transmit History Access Register _m (_m = 0 to 2) ... | 1049 |
| 17.4.15 | Details of Test-related Registers | 1051 |
| 17.4.15.1 | RSCFDnCFDGTSTCFG — Global Test Configuration Register | 1051 |
| 17.4.15.2 | RSCFDnCFDGTSTCTR — Global Test Control Register | 1053 |
| 17.4.15.3 | RSCFDnCFDGLOCKK — Global Lock Key Register | 1054 |
| 17.4.15.4 | RSCFDnCFDRPGACC _r — RAM Test Page Access Register _r (_r = 0 to 63) | 1055 |
| 17.4.16 | Details of Mode Read Register | 1056 |
| 17.4.16.1 | RSCFDnCANFMDR — CAN FD Mode Read Register | 1056 |
| 17.5 | Interrupt Sources and DMA Trigger | 1057 |
| 17.5.1 | Interrupt Sources | 1057 |
| 17.5.2 | DMA Trigger (Only in CAN FD Mode) | 1061 |
| 17.6 | CAN Modes | 1062 |
| 17.6.1 | Global Modes | 1062 |
| 17.6.1.1 | Global Stop Mode | 1064 |
| 17.6.1.2 | Global Reset Mode | 1064 |
| 17.6.1.3 | Global Test Mode | 1064 |
| 17.6.1.4 | Global Operating Mode | 1064 |
| 17.6.2 | Channel Modes | 1065 |
| 17.6.2.1 | Channel Stop Mode | 1066 |
| 17.6.2.2 | Channel Reset Mode | 1066 |
| 17.6.2.3 | Channel Halt Mode | 1067 |
| 17.6.2.4 | Channel Communication Mode | 1068 |
| 17.6.2.5 | Bus Off State | 1069 |
| 17.6.3 | Initializing Registers by Transition to CAN Mode | 1070 |
| 17.7 | Reception Functions | 1072 |
| 17.7.1 | Data Processing Using the Receive Rule Table | 1072 |
| 17.7.1.1 | Acceptance Filter Processing | 1073 |
| 17.7.1.2 | DLC Filter Processing | 1074 |
| 17.7.1.3 | Routing Processing | 1074 |
| 17.7.1.4 | Label Addition Processing | 1074 |
| 17.7.1.5 | Mirror Function Processing | 1074 |
| 17.7.1.6 | Timestamp | 1075 |
| 17.8 | Transmission Functions | 1076 |
| 17.8.1 | Transmit Priority Determination | 1077 |
| 17.8.2 | Transmission Using Transmit Buffers | 1077 |
| 17.8.2.1 | Transmit Abort Function | 1077 |
| 17.8.2.2 | One-Shot Transmission Function (Retransmission Disabling Function) | 1078 |
| 17.8.2.3 | Transmit Buffer Merge Mode (Only in CAN FD Mode) | 1078 |
| 17.8.3 | Transmission Using FIFO Buffers | 1078 |
| 17.8.3.1 | Interval Transmission Function | 1079 |

| | | |
|--|---|-------------|
| 17.8.4 | Transmission Using Transmit Queues | 1082 |
| 17.8.5 | Transmit Data Padding (Only in CAN FD Mode)..... | 1082 |
| 17.8.6 | Transmit History Function..... | 1082 |
| 17.9 | Gateway Function..... | 1084 |
| 17.9.1 | CAN-CAN FD Gateway (Only in CAN FD Mode) | 1084 |
| 17.10 | Test Function | 1085 |
| 17.10.1 | Standard Test Mode | 1085 |
| 17.10.2 | Listen-Only Mode | 1085 |
| 17.10.3 | Self-Test Mode (Loopback Mode) | 1086 |
| 17.10.3.1 | Self-Test Mode 0 (External Loopback Mode)..... | 1086 |
| 17.10.3.2 | Self-Test Mode 1 (Internal Loopback Mode) | 1087 |
| 17.10.4 | Restricted Operation Mode (Only in CAN FD Mode)..... | 1087 |
| 17.10.5 | RAM Test..... | 1087 |
| 17.10.6 | Inter-Channel Communication Test..... | 1088 |
| 17.10.6.1 | CRC Error Test..... | 1089 |
| 17.11 | RS-CANFD Setting Procedure | 1090 |
| 17.11.1 | Initial Settings | 1090 |
| 17.11.1.1 | Clock Setting | 1092 |
| 17.11.1.2 | Bit Timing Setting | 1092 |
| 17.11.1.3 | Communication Speed Setting..... | 1094 |
| 17.11.1.4 | Receive Rule Setting..... | 1096 |
| 17.11.1.5 | Buffer Setting..... | 1097 |
| 17.11.1.6 | Transmitter Delay Compensation (Only in CAN FD Mode)..... | 1099 |
| 17.11.2 | Reception Procedure..... | 1100 |
| 17.11.2.1 | Receive Buffer Reading Procedure..... | 1100 |
| 17.11.2.2 | FIFO Buffer Reading Procedure..... | 1102 |
| 17.11.2.3 | FIFO Buffer Reading Procedure by DMA Transfer | 1106 |
| 17.11.3 | Transmission Procedure..... | 1107 |
| 17.11.3.1 | Procedure for Transmission from Transmit Buffers..... | 1107 |
| 17.11.3.2 | Procedure for Transmission from Transmit/Receive FIFO Buffers | 1112 |
| 17.11.3.3 | Procedure for Transmission from the Transmit Queue | 1116 |
| 17.11.3.4 | Transmit History Buffer Reading Procedure..... | 1117 |
| 17.11.4 | Test Settings..... | 1118 |
| 17.11.4.1 | Self-Test Mode Setting Procedure | 1118 |
| 17.11.4.2 | Procedure for Releasing the Protection | 1119 |
| 17.11.4.3 | RAM Test Setting Procedure..... | 1120 |
| 17.11.4.4 | Inter-Channel Communication Test Setting Procedure..... | 1121 |
| 17.12 | Notes on the RS-CANFD Module | 1122 |
| Section 18 FlexRay (FLXA) | | 1124 |
| 18.1 | Features of RH850/P1M-E FLXA | 1124 |
| 18.1.1 | Number of Units and Channels | 1124 |
| 18.1.2 | Register Base Address..... | 1124 |
| 18.1.3 | Clock Supply..... | 1125 |
| 18.1.4 | Interrupt Requests | 1125 |
| 18.1.5 | Reset Source..... | 1125 |

| | | |
|-----------|---|------|
| 18.1.6 | External Input/Output Pins..... | 1126 |
| 18.1.7 | Combinations of Pins and Ports | 1126 |
| 18.1.8 | Functions..... | 1127 |
| 18.1.9 | Block Diagram | 1128 |
| 18.2 | Register | 1132 |
| 18.2.1 | Register Map | 1132 |
| 18.2.2 | FlexRay Operation register..... | 1135 |
| 18.2.2.1 | FLXAnFROC — FlexRay Operation Control Register..... | 1135 |
| 18.2.2.2 | FLXAnFROS — FlexRay Operation Status Register | 1138 |
| 18.2.3 | Special Registers..... | 1141 |
| 18.2.3.1 | FLXAnFRLCK — FlexRay Lock Register..... | 1141 |
| 18.2.4 | Interrupt Registers | 1142 |
| 18.2.4.1 | FLXAnFREIR — FlexRay Error Interrupt Register | 1142 |
| 18.2.4.2 | FLXAnFRSIR — FlexRay Status Interrupt Register | 1148 |
| 18.2.4.3 | FLXAnFREILS — FlexRay Error Interrupt Line Select Register | 1154 |
| 18.2.4.4 | FLXAnFRSILS — FlexRay Status Interrupt Line Select Register | 1156 |
| 18.2.4.5 | FLXAnFREIES — FlexRay Error Interrupt Enable Set Register | 1158 |
| 18.2.4.6 | FLXAnFREIER — FlexRay Error Interrupt Enable Reset Register | 1160 |
| 18.2.4.7 | FLXAnFRSIES — FlexRay Status Interrupt Enable Set Register | 1162 |
| 18.2.4.8 | FLXAnFRSIER — FlexRay Status Interrupt Enable Reset Register..... | 1164 |
| 18.2.4.9 | FLXAnFRILE — FlexRay Interrupt Line Enable Register..... | 1166 |
| 18.2.5 | FlexRay Timer Registers..... | 1167 |
| 18.2.5.1 | FLXAnFRT0C — FlexRay Timer 0 Configuration Register | 1167 |
| 18.2.5.2 | FLXAnFRT1C — FlexRay Timer 1 Configuration Register | 1169 |
| 18.2.5.3 | FLXAnFRT2C — FlexRay Timer 2 Configuration Register | 1171 |
| 18.2.5.4 | FLXAnFRSTPW1 — FlexRay Stop Watch Register 1 | 1173 |
| 18.2.5.5 | FLXAnFRSTPW2 — FlexRay Stop Watch Register 2 | 1176 |
| 18.2.6 | CC Control Registers..... | 1177 |
| 18.2.6.1 | FLXAnFRSUCC1 — FlexRay SUC Configuration Register 1 | 1177 |
| 18.2.6.2 | FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2 | 1185 |
| 18.2.6.3 | FLXAnFRSUCC3 — FlexRay SUC Configuration Register 3 | 1186 |
| 18.2.6.4 | FLXAnFRNEMC — FlexRay NEM Configuration Register | 1187 |
| 18.2.6.5 | FLXAnFRPRTC1 — FlexRay PRT Configuration Register 1 | 1188 |
| 18.2.6.6 | FLXAnFRPRTC2 — FlexRay PRT Configuration Register 2 | 1191 |
| 18.2.6.7 | FLXAnFRMHDC — FlexRay MHD Configuration Register | 1193 |
| 18.2.6.8 | FLXAnFRGTUC1 — FlexRay GTU Configuration Register 1 | 1194 |
| 18.2.6.9 | FLXAnFRGTUC2 — FlexRay GTU Configuration Register 2 | 1195 |
| 18.2.6.10 | FLXAnFRGTUC3 — FlexRay GTU Configuration Register 3 | 1196 |
| 18.2.6.11 | FLXAnFRGTUC4 — FlexRay GTU Configuration Register 4 | 1198 |
| 18.2.6.12 | FLXAnFRGTUC5 — FlexRay GTU Configuration Register 5 | 1200 |
| 18.2.6.13 | FLXAnFRGTUC6 — FlexRay GTU Configuration Register 6 | 1202 |
| 18.2.6.14 | FLXAnFRGTUC7 — FlexRay GTU Configuration Register 7 | 1203 |
| 18.2.6.15 | FLXAnFRGTUC8 — FlexRay GTU Configuration Register 8 | 1204 |
| 18.2.6.16 | FLXAnFRGTUC9 — FlexRay GTU Configuration Register 9 | 1205 |
| 18.2.6.17 | FLXAnFRGTUC10 — FlexRay GTU Configuration Register 10 | 1207 |
| 18.2.6.18 | FLXAnFRGTUC11 — FlexRay GTU Configuration Register 11 | 1208 |
| 18.2.7 | CC Status Registers | 1210 |
| 18.2.7.1 | FLXAnFRCCSV — FlexRay CC Status Vector Register..... | 1210 |
| 18.2.7.2 | FLXAnFRCCEV — FlexRay CC Error Vector Register..... | 1214 |

| | | |
|-----------|--|------|
| 18.2.7.3 | FLXAnFRSCV — FlexRay Slot Counter Value Register..... | 1216 |
| 18.2.7.4 | FLXAnFRMTCCV — FlexRay Macrotick and Cycle Counter Value Register | 1217 |
| 18.2.7.5 | FLXAnFRRCV — FlexRay Rate Correction Value Register | 1218 |
| 18.2.7.6 | FLXAnFROCV — FlexRay Offset Correction Value Register | 1219 |
| 18.2.7.7 | FLXAnFRSFS — FlexRay Sync Frame Status Register..... | 1220 |
| 18.2.7.8 | FLXAnFRSWNIT — FlexRay Symbol Window and NIT Status Register | 1223 |
| 18.2.7.9 | FLXAnFRACCS — FlexRay Aggregated Channel Status Register..... | 1226 |
| 18.2.7.10 | FLXAnFRESIDm — FlexRay Even Sync ID Register m (m = 1 to 15)..... | 1230 |
| 18.2.7.11 | FLXAnFROSIDm — FlexRay Odd Sync ID Register m (m = 1 to 15)..... | 1232 |
| 18.2.7.12 | FLXAnFRNMVm — FlexRay Network Management Vector Register m (m = 1 to 3)..... | 1234 |
| 18.2.8 | Message Buffer Control Registers..... | 1235 |
| 18.2.8.1 | FLXAnFRMRC — FlexRay Message RAM Configuration Register | 1235 |
| 18.2.8.2 | FLXAnFRFRF — FlexRay FIFO Rejection Filter Register | 1239 |
| 18.2.8.3 | FLXAnFRFRFM — FlexRay FIFO Rejection Filter Mask Register..... | 1241 |
| 18.2.8.4 | FLXAnFRFCL — FlexRay FIFO Critical Level Register..... | 1242 |
| 18.2.9 | Message Buffer Status Registers..... | 1243 |
| 18.2.9.1 | FLXAnFRMHDS — FlexRay Message Handler Status Register | 1243 |
| 18.2.9.2 | FLXAnFRLDTS — FlexRay Last Dynamic Transmit Slot Register | 1246 |
| 18.2.9.3 | FLXAnFRFSR — FlexRay FIFO Status Register | 1247 |
| 18.2.9.4 | FLXAnFRMHDF — FlexRay Message Handler Constraints Flags Register..... | 1249 |
| 18.2.9.5 | FLXAnFRTXRQm — FlexRay Transmission Request m (m = 1 to 4) | 1253 |
| 18.2.9.6 | FLXAnFRNDATm — FlexRay New Data Register m (m = 1 to 4) | 1254 |
| 18.2.9.7 | FLXAnFRMBSCm — FlexRay Message Buffer Status Changed Register m (m = 1 to 4)..... | 1255 |
| 18.2.10 | Input Buffer | 1256 |
| 18.2.10.1 | FLXAnFRWRDSm — FlexRay Write Data Section Register m (m = 1 to 64) | 1257 |
| 18.2.10.2 | FLXAnFRWRHS1 — FlexRay Write Header Section Register 1 | 1258 |
| 18.2.10.3 | FLXAnFRWRHS2 — FlexRay Write Header Section Register 2 | 1261 |
| 18.2.10.4 | FLXAnFRWRHS3 — FlexRay Write Header Section Register 3 | 1262 |
| 18.2.10.5 | FLXAnFRIBCM — FlexRay Input Buffer Command Mask Register..... | 1263 |
| 18.2.10.6 | FLXAnFRIBCR — FlexRay Input Buffer Command Request Register | 1265 |
| 18.2.11 | Output Buffer | 1267 |
| 18.2.11.1 | FLXAnFRRDDSm — FlexRay Read Data Section Register m (m = 1 to 64) | 1267 |
| 18.2.11.2 | FLXAnFRRDHS1 — FlexRay Read Header Section Register 1 | 1268 |
| 18.2.11.3 | FLXAnFRRDHS2 — FlexRay Read Header Section Register 2..... | 1270 |
| 18.2.11.4 | FLXAnFRRDHS3 — FlexRay Read Header Section Register 3..... | 1272 |
| 18.2.11.5 | FLXAnFRMBS — FlexRay Message Buffer Status Register | 1274 |
| 18.2.11.6 | FLXAnFROBCM — FlexRay Output Buffer Command Mask Register | 1279 |
| 18.2.11.7 | FLXAnFROBCR — FlexRay Output Buffer Command Request Register..... | 1281 |
| 18.2.12 | Data Transfer Control Register..... | 1284 |
| 18.2.12.1 | FLXAnFRITC — FlexRay Input Transfer Configuration Register..... | 1284 |
| 18.2.12.2 | FLXAnFROTC — FlexRay Output Transfer Configuration Register | 1286 |
| 18.2.12.3 | FLXAnFRIBA — FlexRay Input Pointer Table Base Address Register | 1289 |
| 18.2.12.4 | FLXAnFRFBA — FlexRay FIFO Pointer Table Base Address Register | 1290 |
| 18.2.12.5 | FLXAnFROBA — FlexRay Output Pointer Table Base Address Register | 1291 |
| 18.2.12.6 | FLXAnFRIQC — FlexRay Input Queue Control Register..... | 1292 |
| 18.2.12.7 | FLXAnFRUIR — FlexRay User Input Transfer Request Register..... | 1293 |
| 18.2.12.8 | FLXAnFRUOR — FlexRay User Output Transfer Request Register | 1294 |
| 18.2.12.9 | FLXAnFRAHBC — FlexRay H-Bus Configuration Register | 1296 |

| | | |
|-----------|--|------|
| 18.2.13 | Data Transfer Status Register | 1297 |
| 18.2.13.1 | FLXAnFRITS — FlexRay Input Transfer Status Register | 1297 |
| 18.2.13.2 | FLXAnFROTS — FlexRay Output Transfer Status Register..... | 1301 |
| 18.2.13.3 | FLXAnFRAES — FlexRay Access Error Status Register | 1306 |
| 18.2.13.4 | FLXAnFRAEA — FlexRay Access Error Address Register | 1309 |
| 18.2.13.5 | FLXAnFRDAm — FlexRay message Data Available Register m (m = 0 to 3) | 1310 |
| 18.3 | Functional Description | 1312 |
| 18.3.1 | FlexRay Module Operation Control | 1312 |
| 18.3.1.1 | FlexRay Module Enable | 1312 |
| 18.3.1.2 | FlexRay Module Disable | 1313 |
| 18.3.2 | Communication Cycle..... | 1314 |
| 18.3.2.1 | Static Segment..... | 1314 |
| 18.3.2.2 | Dynamic Segment..... | 1315 |
| 18.3.2.3 | Symbol Window..... | 1315 |
| 18.3.2.4 | Network Idle Time (NIT) | 1315 |
| 18.3.2.5 | Configuration of NIT Start and Offset Correction Start..... | 1316 |
| 18.3.3 | Communication Modes..... | 1317 |
| 18.3.3.1 | Time-triggered Distributed (TT-D) | 1317 |
| 18.3.4 | Clock Synchronization..... | 1318 |
| 18.3.4.1 | Global Time | 1318 |
| 18.3.4.2 | Local Time..... | 1318 |
| 18.3.4.3 | Synchronization Process..... | 1318 |
| 18.3.5 | Error Handling | 1320 |
| 18.3.5.1 | Clock Correction Failed Counter | 1320 |
| 18.3.5.2 | Passive to Active Counter | 1321 |
| 18.3.5.3 | HALT Command..... | 1321 |
| 18.3.5.4 | FREEZE Command..... | 1321 |
| 18.3.6 | Communication Controller States..... | 1322 |
| 18.3.6.1 | Communication Controller State Diagram..... | 1322 |
| 18.3.6.2 | DEFAULT_CONFIG State..... | 1324 |
| 18.3.6.3 | CONFIG State..... | 1324 |
| 18.3.6.4 | READY State..... | 1325 |
| 18.3.6.5 | WAKEUP State | 1326 |
| 18.3.6.6 | STARTUP State | 1330 |
| 18.3.6.7 | NORMAL_ACTIVE State..... | 1335 |
| 18.3.6.8 | NORMAL_PASSIVE State | 1335 |
| 18.3.6.9 | HALT State..... | 1336 |
| 18.3.7 | Network Management | 1337 |
| 18.3.8 | Filtering and Masking | 1338 |
| 18.3.8.1 | Slot Counter Filtering..... | 1338 |
| 18.3.8.2 | Cycle Counter Filtering..... | 1339 |
| 18.3.8.3 | Channel ID Filtering..... | 1340 |
| 18.3.8.4 | FIFO Filtering | 1340 |
| 18.3.9 | Transmit Process..... | 1341 |
| 18.3.9.1 | Static Segment..... | 1341 |
| 18.3.9.2 | Dynamic Segment..... | 1341 |
| 18.3.9.3 | Transmit Buffers..... | 1341 |
| 18.3.9.4 | Frame Transmission..... | 1342 |
| 18.3.9.5 | Null Frame Transmission | 1343 |

| | | |
|-------------------|---|-------------|
| 18.3.10 | Receive Process..... | 1344 |
| 18.3.10.1 | Dedicated Receive Buffers..... | 1344 |
| 18.3.10.2 | Frame Reception..... | 1344 |
| 18.3.10.3 | Null Frame Reception..... | 1345 |
| 18.3.11 | FIFO Function..... | 1346 |
| 18.3.11.1 | Description | 1346 |
| 18.3.11.2 | Configuration of the FIFO..... | 1347 |
| 18.3.11.3 | Access to the FIFO..... | 1348 |
| 18.3.12 | Message Handling..... | 1349 |
| 18.3.12.1 | Reconfiguration of Message Buffers | 1349 |
| 18.3.12.2 | Host access to Message RAM | 1351 |
| 18.3.12.3 | FlexRay Protocol Controller Access to Message RAM | 1357 |
| 18.3.13 | Message RAM..... | 1358 |
| 18.3.13.1 | Header Partition | 1360 |
| 18.3.13.2 | Data Partition..... | 1364 |
| 18.3.13.3 | Message Data Integrity Check | 1365 |
| 18.3.13.4 | Host Handling of Access Errors | 1367 |
| 18.3.14 | Interrupts | 1368 |
| 18.3.15 | Assignment of FlexRay Configuration Parameters..... | 1369 |
| 18.3.16 | Usage of Data Transfer | 1371 |
| 18.3.16.1 | Input Data Transfer | 1372 |
| 18.3.16.2 | Output Data Transfer..... | 1381 |
| 18.3.16.3 | Data Structure Transfer Scheduling..... | 1391 |
| 18.3.16.4 | Behavior in Case of Data Transfer Access Error | 1392 |
| 18.3.16.5 | Behaviors in Case of RAM Read Errors..... | 1393 |
| Section 19 | Single Edge Nibble Transmission (RSENT) | 1395 |
| 19.1 | Features of RH850/P1M-E RSENT | 1395 |
| 19.1.1 | Number of Channels..... | 1395 |
| 19.1.2 | Register Base Address..... | 1395 |
| 19.1.3 | Clock Supply..... | 1396 |
| 19.1.4 | Interrupt Requests | 1396 |
| 19.1.5 | Reset Sources | 1396 |
| 19.1.6 | External Input/Output Signals..... | 1397 |
| 19.1.7 | Combinations of Pins and Ports | 1397 |
| 19.2 | Functions | 1398 |
| 19.2.1 | Block Diagram | 1399 |
| 19.3 | Registers..... | 1400 |
| 19.3.1 | RSENTnTSPC — RSENT Timestamp Register..... | 1401 |
| 19.3.2 | RSENTnTSC — RSENT Timestamp Counter..... | 1403 |
| 19.3.3 | RSENTnCC — RSENT Communications Configuration Register..... | 1404 |
| 19.3.4 | RSENTnBRP — RSENT Baud Rate Prescaler Register..... | 1407 |
| 19.3.5 | RSENTnIDE — RSENT Interrupt/DMA Enable Register..... | 1409 |
| 19.3.6 | RSENTnMDC — RSENT Mode Control Register | 1412 |
| 19.3.7 | RSENTnSPCT — RSENT SPC Transmission Register..... | 1413 |
| 19.3.8 | RSENTnMST — RSENT Mode Status Register | 1414 |
| 19.3.9 | RSENTnCS — RSENT Communication Status Register | 1416 |

| | | |
|-------------------|---|-------------|
| 19.3.10 | RSENTnCSC — RSENT Communication Status Clear Register..... | 1420 |
| 19.3.11 | RSENTnSRTS — RSENT Slow Channel Receive Timestamp Register..... | 1423 |
| 19.3.12 | RSENTnSRXD — RSENT Slow Channel Receive Data Register | 1424 |
| 19.3.13 | RSENTnCPL — RSENT Calibration Pulse Length Register | 1426 |
| 19.3.14 | RSENTnML — RSENT Message Length Register..... | 1427 |
| 19.3.15 | RSENTnFRTS — RSENT Fast Channel Receive Timestamp Register..... | 1428 |
| 19.3.16 | RSENTnFRXD — RSENT Fast Channel Receive Data Register | 1429 |
| 19.3.17 | RSENTTSSEL — RSENT Timestamp Mode Selection Register | 1431 |
| 19.4 | Modes of Operation | 1433 |
| 19.4.1 | RESET Mode..... | 1434 |
| 19.4.2 | CONFIGURATION Mode | 1434 |
| 19.4.3 | OPERATION IDLE Mode | 1434 |
| 19.4.4 | OPERATION ACTIVE Mode | 1434 |
| 19.4.5 | Register Behavior in Operation Modes..... | 1435 |
| 19.5 | Clock Configuration | 1436 |
| 19.5.1 | Timestamp..... | 1436 |
| 19.5.1.1 | Timestamp Clock Configuration | 1436 |
| 19.5.1.2 | Timestamp Counter Operation | 1436 |
| 19.5.2 | Communication Clock Configuration | 1439 |
| 19.5.2.1 | RX BRP Setting..... | 1439 |
| 19.5.2.2 | RX and SPC Tick Settings | 1439 |
| 19.6 | RSENT Operation..... | 1440 |
| 19.6.1 | Changing Operation Modes..... | 1440 |
| 19.6.2 | Message Reception..... | 1443 |
| 19.6.2.1 | Calibration Pulse Reception..... | 1443 |
| 19.6.2.2 | Data Nibble Reception | 1443 |
| 19.6.2.3 | Fast Channel Message Reception | 1444 |
| 19.6.2.4 | Fast Channel Reception Flow | 1448 |
| 19.6.2.5 | Slow Channel Message Reception | 1449 |
| 19.6.2.6 | Slow Channel Reception Flow | 1451 |
| 19.6.2.7 | DMA Flow..... | 1452 |
| 19.6.2.8 | Error Flagging..... | 1453 |
| 19.7 | SPC Function..... | 1455 |
| 19.7.1 | Multiplexing of the SENTnRX and SENTnSPCO Pin Functions | 1457 |
| 19.8 | Interrupts and Checks..... | 1458 |
| Section 20 | PSI5 | 1460 |
| 20.1 | Features of RH850/P1M-E PSI5..... | 1460 |
| 20.1.1 | Number of Channels..... | 1460 |
| 20.1.2 | Register Base Address..... | 1460 |
| 20.1.3 | Clock Supply..... | 1460 |
| 20.1.4 | Interrupt Requests | 1461 |
| 20.1.5 | Reset Sources | 1461 |
| 20.1.6 | External Input/Output Signals..... | 1461 |
| 20.2 | Functions | 1462 |

| | | |
|---------|--|------|
| 20.2.1 | Block Configuration | 1463 |
| 20.3 | Registers..... | 1464 |
| 20.3.1 | PSI5nCHCTRL — PSI5 Channel Control Register | 1466 |
| 20.3.2 | PSI5nIPTIMERCTRL — PSI5 IP Timer Control Register | 1467 |
| 20.3.3 | PSI5nIPTIMER — PSI5 IP Timer Counter | 1468 |
| 20.3.4 | PSI5nOPMCOMM — PSI5 Operating Mode/Communication Mode Register..... | 1469 |
| 20.3.5 | PSI5nOPMBITRATE — PSI5 Operating-Mode Bit Rate Register..... | 1470 |
| 20.3.6 | PSI5nOPMCYCT — PSI5 Operating-Mode Cycle Time Register | 1471 |
| 20.3.7 | PSI5nPSI5INT — PSI5 Interrupt Status Register..... | 1472 |
| 20.3.8 | PSI5nEMRXDATA — PSI5 Receive Data Emulation Register | 1473 |
| 20.3.9 | PSI5nEMRXDST — PSI5 Receive Data Status Emulation Register..... | 1474 |
| 20.3.10 | PSI5nEMRXDTIM — PSI5 Receive Data IP Timer Emulation Register..... | 1475 |
| 20.3.11 | PSI5nEMRXDFIFO — PSI5 Receive Data FIFO Emulation Register | 1476 |
| 20.3.12 | PSI5nEMRXMRXMSG — PSI5 Receive-Message Receive Message Emulation Register | 1477 |
| 20.3.13 | PSI5nEMRXMRXST — PSI5 Receive-Message Channel Receive Status Emulation Register | 1478 |
| 20.3.14 | PSI5nEMRXMRXTIM — PSI5 Receive-Message Channel Receive Timestamp Emulation Register | 1479 |
| 20.3.15 | PSI5nEMRXMFIFO — PSI5 Receive-Message Channel FIFO Emulation Register | 1480 |
| 20.3.16 | PSI5nTXSETTING — PSI5 Transmission Setting Register | 1481 |
| 20.3.17 | PSI5nSYNCCTRL — PSI5 Synchronization Control Register | 1483 |
| 20.3.18 | PSI5nTXST — PSI5 Transmission Status Register | 1484 |
| 20.3.19 | PSI5nTXSTCLR — PSI5 Transmission Status Clear Register | 1485 |
| 20.3.20 | PSI5nTXSTINTEN — PSI5 Transmission Status Interrupt Enable Register..... | 1486 |
| 20.3.21 | PSI5nTXDCTRL — PSI5 Transmit Data Control Register | 1487 |
| 20.3.22 | PSI5nTXDATA — PSI5 Transmit Data Register | 1488 |
| 20.3.23 | PSI5nRXSPLSET — PSI5 Receive Sampling Setting Register | 1490 |
| 20.3.24 | PSI5nRXSmSET — PSI5 Receive Slot m Setting Register (m = 1 to 8)..... | 1491 |
| 20.3.25 | PSI5nRXDATA — PSI5 Receive Data Register..... | 1492 |
| 20.3.26 | PSI5nRXDST — PSI5 Receive Data Status Register | 1493 |
| 20.3.27 | PSI5nRXDTIM — PSI5 Receive Data IP Timer Register | 1494 |
| 20.3.28 | PSI5nRXDFIFO — PSI5 Receive Data FIFO Register | 1495 |
| 20.3.29 | PSI5nRXMODST — PSI5 Receive Module Status Register | 1496 |
| 20.3.30 | PSI5nRXMODSTCLR — PSI5 Receive Module Status Clear Register | 1497 |
| 20.3.31 | PSI5nRXMODSTINTEN — PSI5 Receive Module Status Interrupt Enable Register..... | 1498 |
| 20.3.32 | PSI5nRXMSET — PSI5 Receive Message Channel Setting Register..... | 1499 |
| 20.3.33 | PSI5nRXMRXMSG — PSI5 Receive-Message Receive Message Register | 1500 |
| 20.3.34 | PSI5nRXMRXST — PSI5 Receive-message Channel Receive Status Register | 1502 |
| 20.3.35 | PSI5nRXMRXTIM — PSI5 Receive-message Channel Receive Timestamp Register.. | 1503 |
| 20.3.36 | PSI5nRXMFIFO — PSI5 Receive-message Channel FIFO Register | 1504 |
| 20.3.37 | PSI5nRXMMST — PSI5 Receive-message Channel Module Status Register | 1505 |
| 20.3.38 | PSI5nRXMMSTCLR— PSI5 Receive-message Channel Module Status Clear Register | 1506 |
| 20.3.39 | PSI5nRXMMSTINTEN — PSI5 Receive-message Channel Module Status Interrupt Enable Register | 1507 |
| 20.3.40 | PSI5TSSEL — PSI5 Timestamp Function Mode Selection Register | 1508 |

| | | |
|--|--|-------------|
| 20.4 | Interrupt | 1509 |
| 20.5 | Operation | 1510 |
| 20.5.1 | Setting Operation Mode..... | 1510 |
| 20.5.1.1 | Setting for Transmit/Receive Mode | 1511 |
| 20.5.1.2 | Setting for Data Reception | 1511 |
| 20.5.1.3 | Initial Setting for Asynchronous Mode..... | 1511 |
| 20.5.1.4 | Initial Setting for Synchronous Mode..... | 1512 |
| 20.5.1.5 | Setting for Serial Message Reception | 1513 |
| 20.5.1.6 | Initial Setting in Asynchronous Mode | 1513 |
| 20.5.1.7 | Initial Setting in Synchronous Mode | 1513 |
| 20.5.1.8 | Setting for Timestamping | 1513 |
| 20.5.2 | Operation Flow | 1515 |
| 20.5.2.1 | Starting Operation | 1515 |
| 20.5.2.2 | Flow of Data Reception | 1516 |
| 20.5.2.3 | Flow of Data Transmission..... | 1517 |
| 20.5.2.4 | Serial Message Reception Flow..... | 1519 |
| 20.5.2.5 | Parity and CRC Errors in Received Data | 1520 |
| 20.5.3 | PAS Compatibility Mode..... | 1521 |
| 20.5.4 | Baud Rate..... | 1521 |
| Section 21 Window Watchdog Timer A (WDTA) | | 1522 |
| 21.1 | Features of RH850/P1M-E WDTA..... | 1522 |
| 21.1.1 | Units and Channels | 1522 |
| 21.1.2 | Register Base Address..... | 1522 |
| 21.1.3 | Clock Supply..... | 1523 |
| 21.1.4 | Interrupt Request..... | 1523 |
| 21.1.5 | Reset Sources | 1523 |
| 21.1.6 | External Input/Output Signals..... | 1523 |
| 21.2 | Overview..... | 1524 |
| 21.2.1 | Functional Overview | 1524 |
| 21.2.2 | Block Diagram | 1526 |
| 21.3 | Registers..... | 1527 |
| 21.3.1 | List of Registers..... | 1527 |
| 21.3.2 | WDTAnWDTE — WDTA Enable Register..... | 1528 |
| 21.3.3 | WDTAnEVAC — WDTA Enable VAC Register | 1529 |
| 21.3.4 | WDTAnREF — WDTA Reference Value Register | 1530 |
| 21.3.5 | WDTAnMD — WDTA Mode Register..... | 1531 |
| 21.4 | Interrupt Sources | 1533 |
| 21.5 | Function | 1534 |
| 21.5.1 | WDTA after Release from the Reset State..... | 1534 |
| 21.5.1.1 | Start Modes | 1534 |
| 21.5.1.2 | Count Clock Selection | 1534 |
| 21.5.1.3 | WDTA Settings after Release From The Reset State,..... | 1535 |
| 21.5.1.4 | Default Start Mode Timing..... | 1536 |
| 21.5.1.5 | Software Trigger Start Mode Timing | 1537 |
| 21.5.2 | WDTA Trigger..... | 1538 |

| | | |
|-------------------|---|-------------|
| 21.5.2.1 | Calculating an Activation Code when the VAC Function is Used..... | 1538 |
| 21.5.3 | WDTA Error Detection..... | 1539 |
| 21.5.4 | 75% Interrupt Request Signals..... | 1540 |
| 21.5.5 | Window Function..... | 1541 |
| Section 22 | OS Timer (OSTM)..... | 1542 |
| 22.1 | Features of RH850/P1M-E OSTM..... | 1542 |
| 22.1.1 | Number of Units..... | 1542 |
| 22.1.2 | Register Base Address..... | 1543 |
| 22.1.3 | Clock Supply..... | 1543 |
| 22.1.4 | Interrupt Requests..... | 1544 |
| 22.1.5 | Reset Sources..... | 1544 |
| 22.1.6 | External Input/Output Signals..... | 1544 |
| 22.2 | Overview..... | 1545 |
| 22.2.1 | Functional Overview..... | 1545 |
| 22.2.2 | Block Diagram..... | 1546 |
| 22.2.3 | Counter clock..... | 1547 |
| 22.2.4 | Output Modes (OSTM0 and OSTM1)..... | 1549 |
| 22.2.5 | Interrupt Requests (INTOSTMn)..... | 1550 |
| 22.3 | Registers..... | 1551 |
| 22.3.1 | List of Registers..... | 1551 |
| 22.3.2 | Details of OSTMn Registers..... | 1552 |
| 22.3.2.1 | OSTMnCMP — OSTMn Compare Register..... | 1552 |
| 22.3.2.2 | OSTMnCNT — OSTMn Counter Register..... | 1553 |
| 22.3.2.3 | OSTMnTO — OSTMn Output Register..... | 1554 |
| 22.3.2.4 | OSTMnTOE — OSTMn Output Enable Register..... | 1554 |
| 22.3.2.5 | OSTMnTE — OSTMn Count Enable Status Register..... | 1555 |
| 22.3.2.6 | OSTMnTS — OSTMn Count Start Trigger Register..... | 1555 |
| 22.3.2.7 | OSTMnTT — OSTMn Count Stop Trigger Register..... | 1556 |
| 22.3.2.8 | OSTMnCTL — OSTMn Control Register..... | 1556 |
| 22.3.2.9 | IC0CKSEL0 — OSTM0 Clock Select Register..... | 1557 |
| 22.3.2.10 | IC0CKSEL1 — OSTM1 Clock Select Register..... | 1559 |
| 22.4 | Operation..... | 1561 |
| 22.4.1 | Starting and Stopping OSTMn..... | 1561 |
| 22.4.2 | Interval Timer Mode..... | 1562 |
| 22.4.2.1 | Basic Operation in Interval Timer Mode..... | 1562 |
| 22.4.2.2 | Operation when OSTMnCMP = 0000 0000 _H | 1565 |
| 22.4.2.3 | Setting Procedure for Interval Timer Mode..... | 1566 |
| 22.4.3 | Free-Running Comparison Mode..... | 1567 |
| 22.4.3.1 | Basic Operation in Free-Running Comparison Mode..... | 1567 |
| 22.4.3.2 | Operation when OSTMnCMP = 0000 0000 _H | 1569 |
| 22.4.3.3 | Setting Procedure for Free-Running Comparison Mode..... | 1570 |
| Section 23 | Timer Array Unit D (TAUD)..... | 1571 |
| 23.1 | Features of RH850/P1M-E TAUD..... | 1571 |
| 23.1.1 | Units and Channels..... | 1571 |

| | | |
|----------|---|------|
| 23.1.2 | Register Base Address..... | 1572 |
| 23.1.3 | Clock Supply..... | 1572 |
| 23.1.4 | Interrupt Requests..... | 1572 |
| 23.1.5 | Reset Sources..... | 1573 |
| 23.1.6 | External Input/Output Signals..... | 1574 |
| 23.1.7 | Internal Input/Output Signals..... | 1574 |
| 23.2 | Overview..... | 1575 |
| 23.2.1 | Functional Overview..... | 1575 |
| 23.2.2 | Terms..... | 1576 |
| 23.2.3 | Functional List of Timer Operations..... | 1577 |
| 23.2.4 | TAUD I/O and Interrupt Request Signals..... | 1578 |
| 23.2.5 | Block Diagram..... | 1579 |
| 23.2.6 | Description of Blocks..... | 1580 |
| 23.3 | Registers..... | 1581 |
| 23.3.1 | List of Registers..... | 1581 |
| 23.3.2 | Details of TAUDn Prescaler Registers..... | 1582 |
| 23.3.2.1 | TAUDnTPS — TAUDn Prescaler Clock Select Register..... | 1582 |
| 23.3.2.2 | TAUDnBRS — TAUDn Prescaler Baud Rate Setting Register..... | 1585 |
| 23.3.3 | Details of TAUDn Control Registers..... | 1586 |
| 23.3.3.1 | TAUDnCDRm — TAUDn Channel Data Register..... | 1586 |
| 23.3.3.2 | TAUDnCNTm — TAUDn Channel Counter Register..... | 1586 |
| 23.3.3.3 | TAUDnCMORM — TAUDn Channel Mode OS Register..... | 1588 |
| 23.3.3.4 | TAUDnCMURm — TAUDn Channel Mode User Register..... | 1591 |
| 23.3.3.5 | TAUDnCSRm — TAUDn channel status register..... | 1592 |
| 23.3.3.6 | TAUDnCSCm - TAUDn Channel Status Clear Register..... | 1593 |
| 23.3.3.7 | TAUDnTS — TAUDn Channel Start Trigger Register..... | 1593 |
| 23.3.3.8 | TAUDnTE — TAUDn Channel Enable Status Register..... | 1594 |
| 23.3.3.9 | TAUDnTT — TAUDn Channel Stop Trigger Register..... | 1594 |
| 23.3.4 | Details of TAUDn Simultaneous Rewrite Registers..... | 1595 |
| 23.3.4.1 | TAUDnRDE — TAUDn Channel Reload Data Enable Register..... | 1595 |
| 23.3.4.2 | TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register..... | 1595 |
| 23.3.4.3 | TAUDnRDM — TAUDn Channel Reload Data Mode Register..... | 1596 |
| 23.3.4.4 | TAUDnRDC — TAUDn Channel Reload Data Control Register..... | 1596 |
| 23.3.4.5 | TAUDnRDT — TAUDn Channel Reload Data Trigger Register..... | 1597 |
| 23.3.4.6 | TAUDnRSF — TAUDn Channel Reload Status Register..... | 1597 |
| 23.3.5 | Details of TAUDn Output Registers..... | 1598 |
| 23.3.5.1 | TAUDnTOE — TAUDn Channel Output Enable Register..... | 1598 |
| 23.3.5.2 | TAUDnTO — TAUDn Channel Output Register..... | 1598 |
| 23.3.5.3 | TAUDnTOM — TAUDn Channel Output Mode Register..... | 1599 |
| 23.3.5.4 | TAUDnTOC — TAUDn Channel Output Configuration Register..... | 1599 |
| 23.3.5.5 | TAUDnTOL — TAUDn Channel Output Active Level Register..... | 1600 |
| 23.3.6 | Details of TAUDn Dead Time Output Registers..... | 1601 |
| 23.3.6.1 | TAUDnTDE — TAUDn Channel Dead Time Output Enable Register..... | 1601 |
| 23.3.6.2 | TAUDnTDM — TAUDn Channel Dead Time Output Mode Register..... | 1601 |
| 23.3.6.3 | TAUDnTDL — TAUDn Channel Dead Time Output Level Register..... | 1602 |
| 23.3.7 | Details of TAUDn Real-time/Modulation Output Registers..... | 1603 |
| 23.3.7.1 | TAUDnTRE — TAUDn Channel Real-time Output Enable Register..... | 1603 |
| 23.3.7.2 | TAUDnTRC — TAUDn Channel Real-time Output Control Register..... | 1603 |

| | | |
|----------|---|------|
| 23.3.7.3 | TAUDnTRO — TAUDn Channel Real-time Output Register..... | 1604 |
| 23.3.7.4 | TAUDnTME — TAUDn Channel Modulation Output Enable Register | 1604 |
| 23.4 | Operating Procedure | 1605 |
| 23.5 | Concepts of Synchronous Channel Operation | 1606 |
| 23.5.1 | Rules of Synchronous Channel Operation | 1606 |
| 23.5.2 | Simultaneous Start and Stop of Synchronous Channel Counters..... | 1608 |
| 23.5.2.1 | Simultaneous Start and Stop within the Same Unit..... | 1608 |
| 23.5.2.2 | Simultaneous Start between the Units | 1608 |
| 23.6 | Simultaneous Rewrite..... | 1609 |
| 23.6.1 | Overview of Operations | 1609 |
| 23.6.2 | How to Control Simultaneous Rewrite..... | 1611 |
| 23.6.2.1 | Initial Settings | 1612 |
| 23.6.2.2 | Start Counter and Count Operation..... | 1612 |
| 23.6.2.3 | Simultaneous Rewrite | 1612 |
| 23.6.3 | Other General Rules of Simultaneous Rewrite..... | 1613 |
| 23.6.4 | Types of Simultaneous Rewrite..... | 1614 |
| 23.6.4.1 | Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)..... | 1614 |
| 23.6.4.2 | Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B)..... | 1616 |
| 23.6.4.3 | Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1)..... | 1618 |
| 23.6.4.4 | Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal (Method C2) | 1620 |
| 23.7 | Channel Output Modes..... | 1622 |
| 23.7.1 | General Procedures for Specifying a Channel Output Mode | 1624 |
| 23.7.2 | Channel Output Modes Controlled Independently by TAUDn Signals | 1625 |
| 23.7.2.1 | Independent Channel Output Mode 1 | 1625 |
| 23.7.2.2 | Independent Channel Output Mode 1 with Real-Time Output | 1625 |
| 23.7.2.3 | Independent Channel Output Mode 2 | 1626 |
| 23.7.3 | Channel Output Modes Controlled Synchronously by TAUDn Signals | 1627 |
| 23.7.3.1 | Synchronous Channel Output Mode 1 | 1627 |
| 23.7.3.2 | Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output..... | 1627 |
| 23.7.3.3 | Synchronous Channel Output Mode 2 | 1628 |
| 23.7.3.4 | Synchronous Channel Output Mode 2 with Dead Time Output | 1628 |
| 23.7.3.5 | Synchronous Channel Output Mode 2 with One-Phase PWM Output | 1629 |
| 23.7.3.6 | Synchronous Channel Output Mode 2 with Complementary Modulation Output | 1630 |
| 23.7.3.7 | Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output..... | 1630 |
| 23.8 | Start Timing in Each Operating Modes..... | 1631 |
| 23.8.1 | Interval Timer Mode, Judge Mode, Capture Mode, and Up/Down Count Mode..... | 1631 |
| 23.8.2 | Event Count Mode..... | 1632 |
| 23.8.3 | Other Operating Modes | 1632 |
| 23.9 | TAUDnTTOUtm Output and INTTAUDnIm Generation when Counter Starts or Restarts | 1633 |
| 23.10 | Interrupt Generation upon Overflow | 1634 |

| | | |
|-----------|---|------|
| 23.10.1 | Combination of TAUDTTINm Input Pulse Interval Measuring Function and TAUDTTINm Input Interval Timer Function..... | 1635 |
| 23.10.2 | Combination of TAUDTTINm Input Signal Width Measuring Function and Overflow Interrupt Output Function (during TAUDTTINm Width Measurement) | 1636 |
| 23.10.3 | Combination of TAUDTTINm Input Position Detecting Function and Interval Timer Function..... | 1637 |
| 23.10.4 | Combination of TAUDTTINm Input Period Count Detecting Function and Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection) | 1638 |
| 23.11 | TAUDnTTINm Edge Detection | 1640 |
| 23.12 | Independent Channel Operation Functions | 1641 |
| 23.12.1 | Interval Timer Function..... | 1641 |
| 23.12.1.1 | Overview | 1641 |
| 23.12.1.2 | Equations | 1641 |
| 23.12.1.3 | Block Diagram and General Timing Diagram..... | 1642 |
| 23.12.1.4 | Register Settings..... | 1643 |
| 23.12.1.5 | Operating Procedure for Interval Timer Function | 1645 |
| 23.12.1.6 | Specific Timing Diagrams..... | 1646 |
| 23.12.2 | TAUDnTTINm Input Interval Timer Function..... | 1651 |
| 23.12.2.1 | Overview | 1651 |
| 23.12.2.2 | Equations | 1651 |
| 23.12.2.3 | Block Diagram and General Timing Diagram..... | 1652 |
| 23.12.2.4 | Operating Procedure for TAUDnTTINm Input Interval Timer Function | 1655 |
| 23.12.2.5 | Specific Timing Diagrams..... | 1656 |
| 23.12.3 | Clock Divider Function..... | 1657 |
| 23.12.3.1 | Overview | 1657 |
| 23.12.3.2 | Equations | 1658 |
| 23.12.3.3 | Block Diagram and General Timing Diagram..... | 1658 |
| 23.12.3.4 | Register Settings..... | 1659 |
| 23.12.3.5 | Operating Procedure for Clock Divider Function..... | 1661 |
| 23.12.3.6 | Specific Timing Diagrams..... | 1662 |
| 23.12.4 | External Event Count Function..... | 1664 |
| 23.12.4.1 | Overview | 1664 |
| 23.12.4.2 | Equations | 1664 |
| 23.12.4.3 | Block Diagram and General Timing Diagram..... | 1665 |
| 23.12.4.4 | Register Settings..... | 1666 |
| 23.12.4.5 | Operating Procedure for External Event Count Function..... | 1667 |
| 23.12.4.6 | Specific Timing Diagrams..... | 1668 |
| 23.12.5 | Delay Count Function..... | 1670 |
| 23.12.5.1 | Overview | 1670 |
| 23.12.5.2 | Equations | 1670 |
| 23.12.5.3 | Block Diagram and General Timing Diagram..... | 1671 |
| 23.12.5.4 | Register Settings..... | 1672 |
| 23.12.5.5 | Operating Procedure for Delay Count Function | 1673 |
| 23.12.6 | One-Pulse Output Function | 1674 |
| 23.12.6.1 | Overview | 1674 |
| 23.12.6.2 | Equations | 1674 |
| 23.12.6.3 | Block Diagram and General Timing Diagram..... | 1675 |
| 23.12.6.4 | Register Settings..... | 1676 |
| 23.12.6.5 | Operating Procedure for One-Pulse Output Function | 1678 |

| | | |
|------------|---|------|
| 23.12.7 | TAUDnTTINm Input Pulse Interval Measurement Function | 1679 |
| 23.12.7.1 | Overview | 1679 |
| 23.12.7.2 | Equations | 1680 |
| 23.12.7.3 | Block Diagram and General Timing Diagram..... | 1681 |
| 23.12.7.4 | Register Settings..... | 1682 |
| 23.12.7.5 | Operating Procedure for TAUDnTTINm Input Pulse Interval Measurement Function..... | 1683 |
| 23.12.7.6 | Specific Timing Diagrams: Overflow Operation..... | 1684 |
| 23.12.8 | TAUDnTTINm Input Signal Width Measurement Function..... | 1687 |
| 23.12.8.1 | Overview | 1687 |
| 23.12.8.2 | Equations | 1688 |
| 23.12.8.3 | Block Diagram and General Timing Diagram..... | 1688 |
| 23.12.8.4 | Register Settings..... | 1690 |
| 23.12.8.5 | Operating Procedure for TAUDnTTINm Input Signal Width Measurement Function..... | 1691 |
| 23.12.8.6 | Specific Timing Diagrams: Overflow Operation..... | 1692 |
| 23.12.9 | TAUDnTTINm Input Position Detection Function | 1696 |
| 23.12.9.1 | Overview | 1696 |
| 23.12.9.2 | Equations | 1696 |
| 23.12.9.3 | Block Diagram and General Timing Diagram..... | 1697 |
| 23.12.9.4 | Register Settings..... | 1698 |
| 23.12.9.5 | Operating Procedure for TAUDnTTINm Input Position Detection Function | 1699 |
| 23.12.9.6 | Specific Timing Diagrams..... | 1700 |
| 23.12.10 | TAUDnTTINm Input Period Count Detection Function..... | 1701 |
| 23.12.10.1 | Overview | 1701 |
| 23.12.10.2 | Equations | 1701 |
| 23.12.10.3 | Block Diagram and General Timing Diagram..... | 1702 |
| 23.12.10.4 | Register Settings..... | 1703 |
| 23.12.10.5 | Operating Procedure for TAUDnTTINm Input Period Count Detection Function | 1704 |
| 23.12.10.6 | Specific Timing Diagrams..... | 1705 |
| 23.12.11 | TAUDnTTINm Input Pulse Interval Judgment Function | 1706 |
| 23.12.11.1 | Overview | 1706 |
| 23.12.11.2 | Block Diagram and General Timing Diagram..... | 1707 |
| 23.12.11.3 | Register Settings..... | 1708 |
| 23.12.11.4 | Operating Procedure for TAUDnTTINm Input Pulse Interval Judgment Function..... | 1709 |
| 23.12.12 | TAUDnTTINm Input Signal Width Judgment Function..... | 1710 |
| 23.12.12.1 | Overview | 1710 |
| 23.12.12.2 | Block Diagram and General Timing Diagram..... | 1711 |
| 23.12.12.3 | Register Settings..... | 1712 |
| 23.12.12.4 | Operating Procedure for TAUDnTTINm Input Signal Width Judgment Function | 1713 |
| 23.12.13 | One-Phase PWM Output Function..... | 1714 |
| 23.12.13.1 | Overview | 1714 |
| 23.12.13.2 | Block Diagram and General Timing Diagram..... | 1715 |
| 23.12.13.3 | Register Settings for Lower Channels..... | 1717 |
| 23.12.13.4 | Register Settings for Upper Channels..... | 1719 |
| 23.12.13.5 | Operating Procedure for One-phase PWM Output Function..... | 1720 |
| 23.12.14 | Real Time Output Function Type 1..... | 1721 |
| 23.12.14.1 | Overview | 1721 |
| 23.12.14.2 | Equations | 1722 |

| | | |
|------------|--|------|
| 23.12.14.3 | Block Diagram and General Timing Diagram..... | 1722 |
| 23.12.14.4 | Register Settings for Upper Channels..... | 1724 |
| 23.12.14.5 | Register Settings for Lower Channels..... | 1726 |
| 23.12.14.6 | Operating Procedure for Real-Time Output Function Type 1..... | 1727 |
| 23.12.14.7 | Specific Timing Diagrams..... | 1728 |
| 23.12.15 | Real-Time Output Function Type 2 | 1729 |
| 23.12.15.1 | Overview | 1729 |
| 23.12.15.2 | Block Diagram and General Timing Diagram..... | 1730 |
| 23.12.15.3 | Register Settings for Upper Channels..... | 1732 |
| 23.12.15.4 | Register Settings for Lower Channels..... | 1734 |
| 23.12.15.5 | Operating Procedure for Real-Time Output Function Type 2..... | 1735 |
| 23.12.15.6 | Specific Timing Diagrams..... | 1736 |
| 23.12.16 | Simultaneous Rewrite Trigger Generation Function Type 1..... | 1737 |
| 23.12.16.1 | Overview | 1737 |
| 23.12.16.2 | Equations | 1738 |
| 23.12.16.3 | Block Diagram and General Timing Diagram..... | 1739 |
| 23.12.16.4 | Register Settings for Upper Channels..... | 1741 |
| 23.12.16.5 | Register Settings for Lower Channels..... | 1742 |
| 23.12.16.6 | Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1 | 1743 |
| 23.13 | Synchronous Channel Operation Functions | 1744 |
| 23.13.1 | PWM Output Function | 1744 |
| 23.13.1.1 | Overview | 1744 |
| 23.13.1.2 | Equations | 1745 |
| 23.13.1.3 | Block Diagram and General Timing Diagram..... | 1746 |
| 23.13.1.4 | Register Settings for the Master Channel | 1748 |
| 23.13.1.5 | Register Settings for Slave Channels..... | 1750 |
| 23.13.1.6 | Operating Procedure for PWM Output Function | 1752 |
| 23.13.1.7 | Specific Timing Diagrams..... | 1753 |
| 23.13.2 | One-Shot Pulse Output Function..... | 1756 |
| 23.13.2.1 | Overview | 1756 |
| 23.13.2.2 | Equations | 1757 |
| 23.13.2.3 | Block Diagram and General Timing Diagram..... | 1757 |
| 23.13.2.4 | Register Settings for the Master Channel | 1760 |
| 23.13.2.5 | Register Settings for Slave Channels..... | 1762 |
| 23.13.2.6 | Operating Procedure for One-Shot Pulse Output Function..... | 1764 |
| 23.13.2.7 | Specific Timing Diagrams..... | 1765 |
| 23.13.3 | Delay Pulse Output Function..... | 1769 |
| 23.13.3.1 | Overview | 1769 |
| 23.13.3.2 | Equations | 1770 |
| 23.13.3.3 | Block Diagram and General Timing Diagram..... | 1771 |
| 23.13.3.4 | Register Settings for the Master Channels..... | 1773 |
| 23.13.3.5 | Register Settings for the Slave Channel 1 | 1775 |
| 23.13.3.6 | Register Settings for Slave Channel 2 | 1777 |
| 23.13.3.7 | Register Settings for Slave Channel 3 | 1779 |
| 23.13.3.8 | Operating Procedure for Delay Pulse Output Function..... | 1781 |
| 23.13.3.9 | Specific Timing Diagrams..... | 1783 |
| 23.13.4 | Offset Trigger Output Function | 1785 |
| 23.13.4.1 | Overview | 1785 |

| | | |
|-----------|--|------|
| 23.13.4.2 | Equations | 1786 |
| 23.13.4.3 | Block Diagram and General Timing Diagram..... | 1786 |
| 23.13.4.4 | Register Settings for Master Channels..... | 1788 |
| 23.13.4.5 | Register Settings for Slave Channels..... | 1790 |
| 23.13.4.6 | Operating Procedure for Offset Trigger Output Function | 1792 |
| 23.13.4.7 | Specific Timing Diagrams..... | 1793 |
| 23.13.5 | A/D Conversion Trigger Output Function Type 1 | 1795 |
| 23.13.5.1 | Overview | 1795 |
| 23.13.5.2 | Block Diagram and General Timing Diagram..... | 1795 |
| 23.13.5.3 | General Timing Diagram | 1796 |
| 23.13.6 | Triangle PWM Output Function | 1797 |
| 23.13.6.1 | Overview | 1797 |
| 23.13.6.2 | Equations | 1798 |
| 23.13.6.3 | Block Diagram and General Timing Diagram..... | 1799 |
| 23.13.6.4 | Register Settings for Master Channels..... | 1801 |
| 23.13.6.5 | Register Settings for Slave Channels..... | 1803 |
| 23.13.6.6 | Operating Procedure for Triangle PWM Output Function | 1805 |
| 23.13.6.7 | Specific Timing Diagrams..... | 1806 |
| 23.13.7 | Triangle PWM Output Function with Dead Time | 1808 |
| 23.13.7.1 | Overview | 1808 |
| 23.13.7.2 | Equations | 1810 |
| 23.13.7.3 | Block Diagram and General Timing Diagram..... | 1811 |
| 23.13.7.4 | Register Settings for Master Channels..... | 1813 |
| 23.13.7.5 | Register Settings for Slave Channel 2 | 1815 |
| 23.13.7.6 | Register Settings for Slave Channel 3 | 1817 |
| 23.13.7.7 | Operating Procedure for Triangle PWM Output Function with Dead Time | 1819 |
| 23.13.7.8 | Specific Timing Diagrams..... | 1820 |
| 23.13.8 | A/D Conversion Trigger Output Function Type 2 | 1822 |
| 23.13.8.1 | Overview | 1822 |
| 23.13.8.2 | Block Diagram and General Timing Diagram..... | 1822 |
| 23.13.9 | Interrupt Request Signals Culling Function | 1824 |
| 23.13.9.1 | Overview | 1824 |
| 23.13.9.2 | Equations | 1825 |
| 23.13.9.3 | Block Diagram and General Timing Diagram..... | 1825 |
| 23.13.9.4 | Register Settings for the Master Channel | 1827 |
| 23.13.9.5 | Register Settings for Slave Channels..... | 1829 |
| 23.13.9.6 | Operating Procedure for Interrupt Request Signals Culling Function | 1830 |
| 23.13.9.7 | Specific Timing Diagram | 1831 |
| 23.14 | Synchronous Non-Complementary and Complementary Modulation Output Functions | 1832 |
| 23.14.1 | Non-Complementary Modulation Output Function Type 1 | 1832 |
| 23.14.1.1 | Overview | 1832 |
| 23.14.1.2 | Equations | 1834 |
| 23.14.1.3 | Block Diagram and General Timing Diagram..... | 1835 |
| 23.14.1.4 | Register Settings for Master Channels..... | 1837 |
| 23.14.1.5 | Register Settings for Slave Channel 1 | 1839 |
| 23.14.1.6 | Register Settings for Slave Channels 2 to 7..... | 1841 |
| 23.14.1.7 | Operating Procedure for Non-Complementary Modulation Output Function Type 1 | 1843 |
| 23.14.1.8 | Specific Timing Diagrams..... | 1845 |

| | | |
|-----------|--|------|
| 23.14.2 | Non-Complementary Modulation Output Function Type 2 | 1846 |
| 23.14.2.1 | Overview | 1846 |
| 23.14.2.2 | Equations | 1848 |
| 23.14.2.3 | Block Diagram and General Timing Diagram..... | 1849 |
| 23.14.2.4 | Register Settings for Master Channels..... | 1851 |
| 23.14.2.5 | Register Settings for Slave Channel 1 | 1853 |
| 23.14.2.6 | Register Settings for slave channels 2 to 7 | 1855 |
| 23.14.2.7 | Operating Procedure for Non-Complementary Modulation Output Function Type 2 | 1857 |
| 23.14.2.8 | Specific Timing Diagrams..... | 1859 |
| 23.14.3 | Complementary Modulation Output Function | 1860 |
| 23.14.3.1 | Overview | 1860 |
| 23.14.3.2 | Equations | 1863 |
| 23.14.3.3 | Block Diagram and General Timing Diagram..... | 1864 |
| 23.14.3.4 | Register Settings for Master Channels..... | 1866 |
| 23.14.3.5 | Register Settings for Slave Channel 1 | 1868 |
| 23.14.3.6 | Register Settings for slave channels 2, 4, and 6 | 1870 |
| 23.14.3.7 | Register Settings for slave channels 3, 5, and 7 | 1872 |
| 23.14.3.8 | Operating Procedure for Complementary Modulation Output Function | 1874 |
| 23.14.3.9 | Specific Timing Diagrams..... | 1876 |

Section 24 Timer Array Unit J (TAUJ)..... 1878

| | | |
|----------|--|------|
| 24.1 | Features of RH850/P1M-E TAUJ | 1878 |
| 24.1.1 | Units and Channels | 1878 |
| 24.1.2 | Register Base Address..... | 1878 |
| 24.1.3 | Clock Supply..... | 1879 |
| 24.1.4 | Interrupt Requests | 1879 |
| 24.1.5 | Reset Sources | 1879 |
| 24.1.6 | External Input/Output Signals..... | 1880 |
| 24.1.7 | Internal Input/Output Signals | 1880 |
| 24.2 | Overview..... | 1881 |
| 24.2.1 | Functional Overview | 1881 |
| 24.2.2 | Terms | 1882 |
| 24.2.3 | Timer Operation Functions | 1883 |
| 24.2.4 | TAUJ I/O and Interrupt Request Signals | 1883 |
| 24.2.5 | Block Diagram | 1884 |
| 24.2.6 | Description of Blocks | 1885 |
| 24.3 | Registers..... | 1886 |
| 24.3.1 | List of Registers..... | 1886 |
| 24.3.2 | Details of TAUJn Prescaler Registers | 1887 |
| 24.3.2.1 | TAUJnTPS — TAUJn Prescaler Clock Select Register | 1887 |
| 24.3.2.2 | TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register..... | 1890 |
| 24.3.3 | Details of TAUJn Control Registers..... | 1891 |
| 24.3.3.1 | TAUJnCDRm — TAUJn Channel Data Register..... | 1891 |
| 24.3.3.2 | TAUJnCNTm — TAUJn Channel Counter Register..... | 1892 |
| 24.3.3.3 | TAUJnCMORm — TAUJn Channel Mode OS Register..... | 1894 |
| 24.3.3.4 | TAUJnCMURm — TAUJn Channel Mode User Register..... | 1897 |

| | | |
|-----------|---|------|
| 24.3.3.5 | TAUJnCSRm — TAUJn Channel Status Register | 1898 |
| 24.3.3.6 | TAUJnCSCm — TAUJn Channel Status Clear Trigger Register | 1898 |
| 24.3.3.7 | TAUJnTS — TAUJn Channel Start Trigger Register | 1899 |
| 24.3.3.8 | TAUJnTE — TAUJn Channel Enable Status Register | 1899 |
| 24.3.3.9 | TAUJnTT — TAUJn Channel Stop Trigger Register | 1900 |
| 24.3.4 | TAUJn Simultaneous Rewrite Register Details | 1901 |
| 24.3.4.1 | TAUJnRDE — TAUJn Channel Reload Data Enable Register | 1901 |
| 24.3.4.2 | TAUJnRDM — TAUJn Channel Reload Data Mode Register | 1901 |
| 24.3.4.3 | TAUJnRDT — TAUJn Channel Reload Data Trigger Register | 1902 |
| 24.3.4.4 | TAUJnRSF — TAUJn Channel Reload Status Register | 1902 |
| 24.3.5 | TAUJn Output Registers Details..... | 1903 |
| 24.3.5.1 | TAUJnTOE — TAUJn Channel Output Enable Register..... | 1903 |
| 24.3.5.2 | TAUJnTO — TAUJn Channel Output Register | 1903 |
| 24.3.5.3 | TAUJnTOM — TAUJn Channel Output Mode Register | 1904 |
| 24.3.5.4 | TAUJnTOC — TAUJn Channel Output Configuration Register | 1905 |
| 24.3.5.5 | TAUJnTOL — TAUJn Channel Output Active Level Register | 1906 |
| 24.4 | Operating Procedure | 1907 |
| 24.5 | Concepts of Synchronous Channel Operation | 1908 |
| 24.5.1 | Rules of Synchronous Channel Operation | 1908 |
| 24.5.2 | Simultaneous Start and Stop of Synchronous Channel Counters..... | 1910 |
| 24.5.2.1 | Simultaneous Start and Stop within a TAUJ Unit | 1910 |
| 24.5.2.2 | Simultaneous Start between TAUJ Units | 1910 |
| 24.6 | Simultaneous Rewrite..... | 1911 |
| 24.6.1 | Functional Overview | 1911 |
| 24.6.2 | How to Control Simultaneous Rewrite..... | 1911 |
| 24.6.2.1 | Initial Settings | 1913 |
| 24.6.2.2 | Start Counter and Count Operation..... | 1913 |
| 24.6.2.3 | Simultaneous Rewrite | 1913 |
| 24.6.3 | Other General Rules for Simultaneous Rewrite | 1913 |
| 24.6.4 | Simultaneous Rewrite Procedure | 1914 |
| 24.7 | Channel Output Modes..... | 1916 |
| 24.7.1 | General Procedures for Specifying a Channel Output Mode | 1918 |
| 24.7.2 | Channel Output Modes Controlled Independently by TAUJn Signals | 1918 |
| 24.7.3 | Channel Output Modes Controlled Synchronously by TAUJn Signals | 1919 |
| 24.8 | Start Timing in Each Operating Modes..... | 1920 |
| 24.8.1 | Interval Timer Mode, Capture Mode, and Count Capture Mode | 1920 |
| 24.8.2 | Other Operating Modes..... | 1921 |
| 24.9 | TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts | 1922 |
| 24.10 | Interrupt Generation upon Overflow | 1923 |
| 24.10.1 | Combination of the TAUJnTTINm Input Position Detection Function and the Interval Timer Function..... | 1924 |
| 24.11 | TAUJnTTINm Edge Detection | 1925 |
| 24.12 | Independent Channel Operation Functions | 1926 |
| 24.12.1 | Interval Timer Function..... | 1926 |
| 24.12.1.1 | Overview | 1926 |
| 24.12.1.2 | Equations | 1926 |

| | | |
|-----------|--|------|
| 24.12.1.3 | Block Diagram and General Timing Diagram..... | 1927 |
| 24.12.1.4 | Register Settings..... | 1928 |
| 24.12.1.5 | Operating Procedure for Interval Timer Function..... | 1930 |
| 24.12.1.6 | Specific Timing Diagrams..... | 1931 |
| 24.12.2 | TAUJnTTINm Input Interval Timer Function..... | 1933 |
| 24.12.2.1 | Overview..... | 1933 |
| 24.12.2.2 | Equations..... | 1933 |
| 24.12.2.3 | Block Diagram and General Timing Diagram..... | 1933 |
| 24.12.2.4 | Register Settings..... | 1935 |
| 24.12.2.5 | Operating Procedure for TAUJnTTINm Input Interval Timer Function..... | 1937 |
| 24.12.2.6 | Specific Timing Diagrams..... | 1938 |
| 24.12.3 | TAUJnTTINm Input Pulse Interval Measurement Function..... | 1939 |
| 24.12.3.1 | Overview..... | 1939 |
| 24.12.3.2 | Equations..... | 1940 |
| 24.12.3.3 | Block Diagram and General Timing Diagram..... | 1940 |
| 24.12.3.4 | Register Settings..... | 1942 |
| 24.12.3.5 | Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function..... | 1943 |
| 24.12.3.6 | Specific Timing Diagrams: Overflow Behavior..... | 1944 |
| 24.12.4 | TAUJnTTINm Input Signal Width Measurement Function..... | 1948 |
| 24.12.4.1 | Overview..... | 1948 |
| 24.12.4.2 | Equations..... | 1949 |
| 24.12.4.3 | Block Diagram and General Timing Diagram..... | 1949 |
| 24.12.4.4 | Register Settings..... | 1951 |
| 24.12.4.5 | Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function..... | 1952 |
| 24.12.4.6 | Specific Timing Diagrams: Overflow Behavior..... | 1953 |
| 24.12.5 | TAUJnTTINm Input Position Detection Function..... | 1957 |
| 24.12.5.1 | Overview..... | 1957 |
| 24.12.5.2 | Equations..... | 1957 |
| 24.12.5.3 | Block Diagram and General Timing Diagram..... | 1958 |
| 24.12.5.4 | Register Settings..... | 1959 |
| 24.12.5.5 | Operating Procedure for TAUJnTTINm Input Position Detection Function..... | 1960 |
| 24.12.5.6 | Specific Timing Diagrams..... | 1961 |
| 24.12.6 | TAUJnTTINm Input Period Count Detection Function..... | 1962 |
| 24.12.6.1 | Overview..... | 1962 |
| 24.12.6.2 | Equations..... | 1962 |
| 24.12.6.3 | Block Diagram and General Timing Diagram..... | 1963 |
| 24.12.6.4 | Register Settings..... | 1964 |
| 24.12.6.5 | Operating Procedure for TAUJnTTINm Input Period Count Detection Function..... | 1965 |
| 24.12.6.6 | Specific Timing Diagrams..... | 1966 |
| 24.13 | Synchronous Channel Operation Functions..... | 1967 |
| 24.13.1 | PWM output function..... | 1967 |
| 24.13.1.1 | Overview..... | 1967 |
| 24.13.1.2 | Equations..... | 1969 |
| 24.13.1.3 | Block Diagram and General Timing Diagram..... | 1969 |
| 24.13.1.4 | Register Settings for the Master Channel..... | 1971 |
| 24.13.1.5 | Register Settings for the Slave Channel(s)..... | 1973 |
| 24.13.1.6 | Operating Procedure for PWM Output Function..... | 1975 |

| | | |
|-------------------|---|-------------|
| 24.13.1.7 | Specific Timing Diagrams..... | 1976 |
| Section 25 | Motor Control Timer (TSG3) | 1979 |
| 25.1 | Features of RH850/P1M-E TSG3..... | 1979 |
| 25.1.1 | Number of Units..... | 1979 |
| 25.1.2 | Register Base Address..... | 1979 |
| 25.1.3 | Clock Supply..... | 1979 |
| 25.1.4 | Interrupt Requests | 1980 |
| 25.1.5 | Reset Sources | 1981 |
| 25.1.6 | External Input/Output Signals..... | 1981 |
| 25.2 | Overview..... | 1982 |
| 25.2.1 | Functional Overview | 1982 |
| 25.2.2 | Block Diagram | 1984 |
| 25.3 | Registers..... | 1985 |
| 25.3.1 | List of Registers..... | 1985 |
| 25.3.2 | TSG3nCTL0 — TSG3n Control Register 0 | 1988 |
| 25.3.3 | TSG3nCTL1 — TSG3n Control Register 1 | 1989 |
| 25.3.4 | TSG3nCTL2 — TSG3n Control Register 2 | 1991 |
| 25.3.5 | TSG3nCTL3 — TSG3n Control Register 3 | 1992 |
| 25.3.6 | TSG3nCTL4 — TSG3n Control Register 4 | 1993 |
| 25.3.7 | TSG3nCTL5 — TSG3n Control Register5 | 1995 |
| 25.3.8 | TSG3nCTL6 — TSG3n Control Register 6 | 1998 |
| 25.3.9 | TSG3nCTL7 — TSG3n Control Register 7 | 2001 |
| 25.3.10 | TSG3nCTL8 — TSG3n Control Register 8 | 2002 |
| 25.3.11 | TSG3nIOC0 — TSG3n I/O Control Register0 | 2003 |
| 25.3.12 | TSG3nIOC1 — TSG3n I/O Control Register1 | 2004 |
| 25.3.13 | TSG3nIOC2 — TSG3n I/O Control Register2 | 2005 |
| 25.3.14 | TSG3nIOC3 — TSG3n I/O Control Register3 | 2006 |
| 25.3.15 | TSG3nSTR0 — TSG3n Status Register 0 | 2007 |
| 25.3.16 | TSG3nSTR1 — TSG3n Status Register 1 | 2008 |
| 25.3.17 | TSG3nSTR2 — TSG3n Status Register 2 | 2009 |
| 25.3.18 | TSG3nSTC — TSG3n Status Clear Trigger Register | 2012 |
| 25.3.19 | TSG3nOPT0 — TSG3n Option Register 0..... | 2014 |
| 25.3.20 | TSG3nOPT1 — TSG3n Option Register 1..... | 2016 |
| 25.3.21 | TSG3nTRG0 — TSG3n Trigger Register 0..... | 2017 |
| 25.3.22 | TSG3nTRG1 — TSG3n Trigger Register 1..... | 2017 |
| 25.3.23 | TSG3nTRG2 — TSG3n Trigger Register 2..... | 2018 |
| 25.3.24 | TSG3nCNT — TSG3n Counter Read Buffer Register | 2018 |
| 25.3.25 | TSG3nCNTE — TSG3n Bit Extended Counter Read Buffer Register..... | 2019 |
| 25.3.26 | TSG3nSBC — TSG3n Sub-Counter Read Buffer Register | 2020 |
| 25.3.27 | TSG3nSBCE — TSG3n Bit Extended Sub-Counter Read Buffer Register | 2020 |
| 25.3.28 | TSG3nCMP0 — TSG3n Compare Register 0..... | 2021 |
| 25.3.29 | TSG3nCMP0E — TSG3n Bit Extended Compare Register 0 | 2021 |
| 25.3.30 | TSG3nCMP1W — TSG3n Compare Register 1, 2..... | 2022 |
| 25.3.31 | TSG3nCMP3W — TSG3n Compare Register 3, 4..... | 2022 |
| 25.3.32 | TSG3nCMP5W — TSG3n Compare Register 5, 6..... | 2023 |

| | | |
|----------|--|------|
| 25.3.33 | TSG3nCMP7W — TSG3n Compare Registers 7, 8..... | 2023 |
| 25.3.34 | TSG3nCMP9W — TSG3n Compare Registers 9, 10..... | 2024 |
| 25.3.35 | TSG3nCMP11W — TSG3n Compare Registers 11, 12..... | 2024 |
| 25.3.36 | TSG3nCMP1 to TSG3nCMP12 — TSG3n Compare Registers 1 to 12..... | 2025 |
| 25.3.37 | TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12..... | 2026 |
| 25.3.38 | TSG3nDCMP0W — TSG3n Diagnostic Output Compare Register 0, 1 | 2028 |
| 25.3.39 | TSG3nDCMP2 — TSG3n Diagnostic Output Compare Register 2..... | 2028 |
| 25.3.40 | TSG3nDCMP0E to 2E — TSG3n Bit Extended Diagnostic Output Compare Register 0 to 2 | 2029 |
| 25.3.41 | TSG3nPAT0W — TSG3n Pattern Register 0..... | 2030 |
| 25.3.42 | TSG3nPAT1W — TSG3n Pattern Register 1..... | 2031 |
| 25.3.43 | TSG3nDTC0W — TSG3n Dead Time Control Register 0..... | 2032 |
| 25.3.44 | TSG3nDTC1W — TSG3n Dead Time Control Register 1..... | 2032 |
| 25.3.45 | TSG3nCMPU — TSG3n HT-PWM U Phase Compare Register..... | 2033 |
| 25.3.46 | TSG3nCMPV — TSG3n HT-PWM V Phase Compare Register | 2033 |
| 25.3.47 | TSG3nCMPW — TSG3n HT-PWM W Phase Compare Register | 2033 |
| 25.3.48 | TSG3nCMPUE — TSG3n Bit Extended HT-PWM U Phase Compare Register | 2034 |
| 25.3.49 | TSG3nCMPVE — TSG3n Bit Extended HT-PWM V Phase Compare Register | 2035 |
| 25.3.50 | TSG3nCMPWE — TSG3n Bit Extended HT-PWM W Phase Compare Register | 2036 |
| 25.3.51 | TSG3nUPW — TSG3n SP-PWM U Phase Active Width Register..... | 2037 |
| 25.3.52 | TSG3nVPW — TSG3n SP-PWM V Phase Active Width Register | 2037 |
| 25.3.53 | TSG3nWPW — TSG3n SP-PWM W Phase Active Width Register | 2037 |
| 25.3.54 | TSG3nUPWE — TSG3n Bit Extended SP-PWM U Phase Active Width Register..... | 2038 |
| 25.3.55 | TSG3nVPWE — TSG3n Bit Extended SP-PWM V Phase Active Width Register | 2039 |
| 25.3.56 | TSG3nWPWE — TSG3n Bit Extended SP-PWM W Phase Active Width Register | 2040 |
| 25.3.57 | TSG3nHSPCMUE — TSG3n HSP-PWM Mode U Phase Compare Register..... | 2041 |
| 25.3.58 | TSG3nHSPCMVE — TSG3n HSP-PWM Mode V Phase Compare Register | 2041 |
| 25.3.59 | TSG3nHSPCMWE — TSG3n HSP-PWM Mode W Phase Compare Register | 2042 |
| 25.3.60 | TSG3nHSPSHUE — TSG3n HSP-PWM Mode U Phase Shift Register..... | 2042 |
| 25.3.61 | TSG3nHSPSHVE — TSG3n HSP-PWM Mode V Phase Shift Mode Register | 2043 |
| 25.3.62 | TSG3nHSPSHWE — TSG3n HSP-PWM Mode W Phase Shift Register | 2043 |
| 25.3.63 | TSG3nDTPR — TSG3n Dead Time Protection Register | 2044 |
| 25.4 | Function..... | 2045 |
| 25.4.1 | Basic Operation..... | 2045 |
| 25.4.1.1 | Basic Operation of 18-Bit Counter..... | 2045 |
| 25.4.1.2 | Function of Compare Registers..... | 2047 |
| 25.4.1.3 | Compare Register Rewrite Operation | 2049 |
| 25.4.1.4 | List of Outputs in Each Mode | 2057 |
| 25.4.1.5 | Restart..... | 2060 |
| 25.4.2 | Match Interrupt | 2061 |
| 25.4.3 | Flags..... | 2066 |
| 25.4.3.1 | Up Count Flag (TSG3nCUF and TSG3nSUF) | 2067 |
| 25.4.3.2 | Positive Phase and Inverse Phase Simultaneous Active State Detection Flag (TSG3nTBF0 to TSG3nTBF2)..... | 2069 |
| 25.4.3.3 | Reload Request Flag (TSG3nRSF)..... | 2070 |
| 25.4.3.4 | Noise Detection Flag (TSG3nNDF)..... | 2071 |

| | | |
|-------------------|---|-------------|
| 25.4.3.5 | Pattern Order Detection Flag (TSG3nTSF)..... | 2072 |
| 25.4.3.6 | Pattern Error Detection Flag (TSG3nPEF)..... | 2074 |
| 25.4.3.7 | Pattern Reversal Detection Flag (TSG3nPRF) | 2075 |
| 25.4.3.8 | TSG3nPTSI2 to TSG3nPTSI0 Pin Abnormal Toggle Detection Flag (TSG3nPTF)..... | 2077 |
| 25.4.3.9 | TSG3nOPCI0 and TSG3nOPCI1 Signal Simultaneous Trigger Detection Flag (TSG3nTDF)..... | 2078 |
| 25.4.3.10 | Pattern Phase Difference Detection Flag (TSG3nPPF) | 2079 |
| 25.4.3.11 | Timer Output Pattern Flag (TSG3nOPF2-TSG3nOPF0)..... | 2080 |
| 25.4.3.12 | Pattern Switch Detection Signal (TSG3nPTE) | 2081 |
| 25.4.4 | Interrupt Skipping Function..... | 2083 |
| 25.4.4.1 | Operation of Interrupt Skipping Function..... | 2084 |
| 25.4.4.2 | Example of Operation when Peak Interrupt is Generated (in PWM Mode)..... | 2087 |
| 25.4.5 | A/D Conversion Trigger Function | 2088 |
| 25.4.5.1 | Operation of A/D Conversion Trigger | 2089 |
| 25.4.6 | Error/Warning Interrupt..... | 2094 |
| 25.4.6.1 | Error Interrupt Function | 2094 |
| 25.4.6.2 | Warning Interrupt Function..... | 2097 |
| 25.4.7 | Operating Modes | 2098 |
| 25.4.7.1 | PWM Mode..... | 2098 |
| 25.4.7.2 | HT-PWM mode (High accuracy Triangular - Pulse Width Modulation mode) | 2109 |
| 25.4.7.3 | SP-PWM Mode (Shifted-pulse - Pulse Width Modulation Mode) | 2131 |
| 25.4.7.4 | 120-DC Mode | 2142 |
| 25.4.7.5 | HSP-PWM Mode (High accuracy Shifted-pulse - Pulse Width Modulation Mode) | 2176 |
| 25.4.7.6 | Compare Register Setting in HSP-PWM Mode..... | 2184 |
| 25.4.7.7 | Timer Output Operation in HSP-PWM Mode | 2185 |
| 25.4.7.8 | Software Output Control Function | 2196 |
| Section 26 | Timer Option (TAPA) | 2197 |
| 26.1 | Features of RH850/P1M-E TAPA | 2197 |
| 26.1.1 | Number of Units..... | 2197 |
| 26.1.2 | Register Base Address..... | 2197 |
| 26.1.3 | Clock Supply..... | 2197 |
| 26.1.4 | Interrupt Requests | 2198 |
| 26.1.5 | Reset Sources | 2198 |
| 26.1.6 | Internal Signal..... | 2198 |
| 26.1.7 | Peripheral Configuration..... | 2199 |
| 26.2 | Overview..... | 2200 |
| 26.2.1 | Functional Overview | 2200 |
| 26.2.2 | Block Diagram | 2200 |
| 26.3 | Registers..... | 2200 |
| 26.3.1 | List of Registers..... | 2200 |
| 26.3.2 | TAPAnCTL0 — TAPAn Control Register 0 | 2201 |
| 26.3.3 | TAPAnFLG — TAPAn Flag Register | 2202 |
| 26.3.4 | TAPAnACWE — TAPAn Asynchronous Control Write Enable Register | 2203 |
| 26.3.5 | TAPAnACTS — TAPAn Asynchronous Control Start Trigger Register | 2204 |

| | | |
|--|--|-------------|
| 26.3.6 | TAPAnACTT — TAPAn Asynchronous Control Stop Trigger Register | 2204 |
| 26.3.7 | TAPAnOPHS — TAPAn Hi-Z Start Trigger Register..... | 2205 |
| 26.3.8 | TAPAnOPHT — TAPAn Hi-Z Stop Trigger Register | 2205 |
| 26.4 | Function | 2206 |
| 26.4.1 | Asynchronous Hi-Z Control Function..... | 2206 |
| 26.4.1.1 | Overview | 2206 |
| 26.4.1.2 | Example of System Configuration..... | 2206 |
| 26.4.1.3 | Basic Operation..... | 2208 |
| 26.4.1.4 | Asynchronous Hi-Z Control by Software Trigger..... | 2210 |
| 26.4.1.5 | Operating Procedure | 2211 |
| Section 27 Timer Pattern Buffer (TPBA)..... | | 2212 |
| 27.1 | Features of RH850/P1M-E TPBA | 2212 |
| 27.1.1 | Units and Channels | 2212 |
| 27.1.2 | Register Base Address..... | 2212 |
| 27.1.3 | Clock Supply..... | 2212 |
| 27.1.4 | Interrupt Requests | 2213 |
| 27.1.5 | Reset Sources | 2213 |
| 27.1.6 | External Input/Output Signals..... | 2213 |
| 27.2 | Overview | 2214 |
| 27.2.1 | Functional Overview | 2214 |
| 27.2.2 | Block Diagram | 2215 |
| 27.3 | Registers..... | 2216 |
| 27.3.1 | List of Registers..... | 2216 |
| 27.3.2 | TPBAnCTL — TPBAn Control Register | 2217 |
| 27.3.3 | TPBAnRDM — TPBAn Reload Data Mode Register..... | 2218 |
| 27.3.4 | TPBAnRSF — TPBAn Reload Status Register | 2219 |
| 27.3.5 | TPBAnRDT — TPBAn Reload Data Trigger Register | 2220 |
| 27.3.6 | TPBAnTOE — TPBAn Timer Output Enable Register | 2220 |
| 27.3.7 | TPBAnTO — TPBAn Timer Output Register..... | 2221 |
| 27.3.8 | TPBAnTOL — TPBAn Timer Output Level Register | 2222 |
| 27.3.9 | TPBAnCMP0 — TPBAn Period Setting Register | 2223 |
| 27.3.10 | TPBAnBUFm — TPBAn Duty Setting Register | 2224 |
| 27.3.11 | TPBAnCMP1 — TPBAn Pattern Number Setting Register | 2225 |
| 27.3.12 | TPBAnCNT0 — TPBAn Timer Counter Register | 2226 |
| 27.3.13 | TPBAnCNT1 — TPBAn Address Counter Register | 2226 |
| 27.3.14 | TPBAnTE — TPBAn Enable Status Register..... | 2227 |
| 27.3.15 | TPBAnTS — TPBAn Start Trigger Register | 2227 |
| 27.3.16 | TPBAnTT — TPBAn Stop Trigger Register..... | 2228 |
| 27.4 | Function | 2229 |
| 27.4.1 | Basic Operation | 2229 |
| 27.4.1.1 | Basic Operation of 16-Bit Counter (TPBAnCNT0) | 2229 |
| 27.4.1.2 | Basic Operation of 7-Bit Counter (TPBAnCNT1) | 2229 |
| 27.4.2 | Compare Register Rewrite Operation | 2230 |
| 27.4.3 | Duty Rewrite Operation | 2233 |
| 27.4.3.1 | TPBAnBUFm Setting Flow | 2233 |

| | | |
|-------------------|---|-------------|
| 27.4.3.2 | Access to TPBAnBUFm | 2234 |
| 27.4.3.3 | Relationship between TPBAnCNT1 Read Value and TPBAnBUFm..... | 2235 |
| 27.4.4 | Basic Operation Example | 2236 |
| 27.4.4.1 | List of Operations | 2237 |
| Section 28 | Encoder Timer (ENCA) | 2241 |
| 28.1 | Features of RH850/P1M-E ENCA | 2241 |
| 28.1.1 | Units and Channels | 2241 |
| 28.1.2 | Register Base Address..... | 2241 |
| 28.1.3 | Clock Supply..... | 2242 |
| 28.1.4 | Interrupt Requests | 2242 |
| 28.1.5 | Reset Sources | 2242 |
| 28.1.6 | External Input/Output Signals..... | 2243 |
| 28.1.7 | Internal Input/Output Signals | 2243 |
| 28.2 | Overview..... | 2244 |
| 28.2.1 | Functional Overview | 2244 |
| 28.2.2 | Block Diagram | 2245 |
| 28.3 | Registers..... | 2246 |
| 28.3.1 | List of Registers..... | 2246 |
| 28.3.2 | ENCA _n CTL — ENCA Control Register | 2247 |
| 28.3.3 | ENCA _n IOC0 — ENCA I/O Control Register 0..... | 2249 |
| 28.3.4 | ENCA _n IOC1 — ENCA I/O Control Register 1 | 2250 |
| 28.3.5 | ENCA _n FLG — ENCA Status Flag Register | 2251 |
| 28.3.6 | ENCA _n FGC — ENCA Status Flag Clear Register | 2252 |
| 28.3.7 | ENCA _n CCR0 — ENCA Capture/Compare Register 0 | 2253 |
| 28.3.8 | ENCA _n CCR1 — ENCA Capture/Compare Register 1 | 2254 |
| 28.3.9 | ENCA _n CNT — ENCA Counter Register | 2255 |
| 28.3.10 | ENCA _n TE — ENCA Timer Enable Status Register | 2256 |
| 28.3.11 | ENCA _n TS — ENCA Timer Start Trigger Register..... | 2257 |
| 28.3.12 | ENCA _n TT — ENCA Timer Stop Trigger Register | 2258 |
| 28.4 | Operation..... | 2259 |
| 28.4.1 | Timer Counter Operation..... | 2259 |
| 28.4.2 | Up/Down Control of Timer Counter | 2261 |
| 28.4.2.1 | When ENCA _n UDS[1:0] Bits in ENCA _n CTL = 00 _B | 2261 |
| 28.4.2.2 | When ENCA _n UDS[1:0] Bits in ENCA _n CTL = 01 _B | 2262 |
| 28.4.2.3 | When ENCA _n UDS[1:0] Bits in ENCA _n CTL = 10 _B | 2263 |
| 28.4.2.4 | When ENCA _n UDS[1:0] Bits in ENCA _n CTL = 11 _B | 2264 |
| 28.4.3 | Timer Counter Clear Control by Encoder Input | 2265 |
| 28.4.3.1 | Clearing Method when ENCA _n SCE = 0..... | 2265 |
| 28.4.3.2 | Clearing Method when ENCA _n SCE = 1 | 2266 |
| 28.4.4 | Functions of ENCA _n CCR0 | 2267 |
| 28.4.4.1 | Compare Function..... | 2267 |
| 28.4.4.2 | Capture Function..... | 2267 |
| 28.4.5 | Functions of ENCA _n CCR1 | 2268 |
| 28.4.5.1 | Compare Function..... | 2268 |
| 28.4.5.2 | Capture Function..... | 2270 |

| | | |
|-------------------|--|-------------|
| 28.4.5.3 | Timer Counter Clearing upon Compare Register Match | 2271 |
| 28.4.6 | Startup/Stop of Timer Counter..... | 2272 |
| 28.4.6.1 | Startup of Timer..... | 2272 |
| 28.4.6.2 | Stop of Timer..... | 2272 |
| 28.5 | ENCA Setting Sequences..... | 2273 |
| 28.5.1 | Encoder timer setting procedure | 2273 |
| 28.5.1.1 | Initial Setting Procedure for Counter | 2273 |
| 28.5.1.2 | Initial Setting Procedure for Counter Clear..... | 2274 |
| 28.5.1.3 | Setting Procedure for ENCAAnCCR0 Register | 2274 |
| 28.5.1.4 | Setting Procedure for ENCAAnCCR1 Register | 2275 |
| 28.6 | Timing Chart | 2276 |
| 28.6.1 | Overflow Occurrence and Overflow Flag Clear Operation | 2276 |
| 28.6.2 | Underflow Occurrence and Underflow Flag Clear Operation | 2277 |
| 28.6.3 | Count Clear and Capture Operation by Encoder Clear Input (ENCAAnEC pin)..... | 2278 |
| 28.6.4 | Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin)..... | 2279 |
| 28.6.5 | Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin)..... | 2280 |
| 28.6.6 | Overflow Operation Immediately after Startup | 2281 |
| 28.6.7 | Underflow Operation Immediately after Startup | 2282 |
| 28.6.8 | Using the ENCAAnLDE Function Immediately after Startup | 2283 |
| 28.6.9 | ENCAAnLDE Function (Loading Count Value)..... | 2284 |
| 28.6.10 | Conflict between ENCAAnLDE Function (Loading Counter Value) and Rewrite of ENCAAnCCR0 Register | 2286 |
| 28.6.11 | Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input..... | 2287 |
| 28.6.12 | Up-count after Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input | 2289 |
| 28.6.13 | Capture Operation between Count Clocks (ENCAAnCCR1)..... | 2290 |
| 28.6.14 | Capture Operation between Count Clocks (ENCAAnCCR0)..... | 2291 |
| 28.6.15 | Encoder operation when compare match clear control is enabled and ENCAAnCTS = 0 | 2292 |
| 28.6.16 | Encoder operation when compare match clear control is enabled and ENCAAnCTS = 1 | 2293 |
| 28.6.17 | Encoder operation when compare match clear control is disabled | 2294 |
| 28.6.18 | Capture Operation Performed upon Clearing by ENCAAnEC, ENCAAnE0, and ENCAAnE1 when ENCAAnSCE = 1 | 2295 |
| 28.6.18.1 | Accompanying Capture Operation | 2295 |
| 28.6.18.2 | When the Timing of the ENCAAnEC Input is Later than that of the ENCAAnE1 Input during Up-count..... | 2296 |
| 28.6.18.3 | When the Timing of the ENCAAnEC Input is the Same as that of the ENCAAnE1 Input during Up-count..... | 2297 |
| 28.6.18.4 | When the Timing of the ENCAAnEC Input is Earlier than that of the ENCAAnE1 Input during Up-count..... | 2297 |
| 28.6.18.5 | When the Timing of the ENCAAnEC Input is Later than that of the ENCAAnE1 Input during Down-count | 2298 |
| 28.6.19 | Capture Operation Performed upon Clearing by ENCAAnEC when ENCAAnSCE = 0..... | 2299 |
| Section 29 | Peripheral Interconnection (PIC)..... | 2300 |
| 29.1 | Features of RH850/P1M-E PIC | 2300 |

| | | |
|-----------|---|------|
| 29.1.1 | Number of Units..... | 2300 |
| 29.1.2 | Register Base Address..... | 2300 |
| 29.1.3 | Clock Supply..... | 2301 |
| 29.1.4 | Reset Sources..... | 2301 |
| 29.1.5 | Input/Output Signals..... | 2301 |
| 29.1.6 | External Input/Output Signals..... | 2303 |
| 29.2 | Peripheral Interconnection — 1 (PIC1A) | 2305 |
| 29.2.1 | Overview..... | 2305 |
| 29.2.1.1 | Functional Overview..... | 2305 |
| 29.2.2 | Registers | 2306 |
| 29.2.2.1 | List of Registers..... | 2306 |
| 29.2.2.2 | PIC1ASST — Simultaneous Start Trigger Control Register..... | 2308 |
| 29.2.2.3 | PIC1ASSER0 — Simultaneous Start Control Register 0 | 2309 |
| 29.2.2.4 | PIC1ASSER1 — Simultaneous Start Control Register 1 | 2309 |
| 29.2.2.5 | PIC1ASSER2 — Simultaneous Start Control Register 2 | 2310 |
| 29.2.2.6 | PIC1ASSER3 — Simultaneous Start Control Register 3 | 2312 |
| 29.2.2.7 | PIC1AINIn0 — Flip-Flop Circuit Initialization Register n0 | 2312 |
| 29.2.2.8 | PIC1AINIn1 — DT Initialization Register n1..... | 2313 |
| 29.2.2.9 | PIC1ATSGHALLSEL — Hall Sensor Input Select Register | 2314 |
| 29.2.2.10 | PIC1ATAUD0SEL — TAUD0 Input Select Register..... | 2315 |
| 29.2.2.11 | PIC1ATAUD1SEL — TAUD1 Input Select Register..... | 2316 |
| 29.2.2.12 | PIC1AHIZCEN0 — Hi-Z Control Register 0 | 2317 |
| 29.2.2.13 | PIC1AHIZCEN1 — Hi-Z Control Register 1 | 2318 |
| 29.2.2.14 | PIC1AHIZCEN2 — Hi-Z Control Register 2 | 2319 |
| 29.2.2.15 | PIC1AHIZCEN3 — Hi-Z Control Register 3 | 2320 |
| 29.2.2.16 | PIC1AENCSEL400 — ENCATIN1 Input Select Register 400..... | 2321 |
| 29.2.2.17 | PIC1AENCSEL410 — ENCATIN1 Input Select Register 410..... | 2322 |
| 29.2.2.18 | PIC1AREG200 — Timer Input/Output Control Register 200..... | 2323 |
| 29.2.2.19 | PIC1AREG210 — Timer Input/Output Control Register 210..... | 2324 |
| 29.2.2.20 | PIC1AREG2n1 — Timer Input/Output Control Register 2n1..... | 2325 |
| 29.2.2.21 | PIC1AREG2n2 — Timer Input/Output Control Register 2n2..... | 2327 |
| 29.2.2.22 | PIC1AREG2n3 — Timer Input/Output Control Register 2n3..... | 2328 |
| 29.2.2.23 | PIC1AREG30 — Timer Input/Output Control Register 30..... | 2330 |
| 29.2.2.24 | PIC1AREG31 — Timer Input/Output Control Register 31..... | 2332 |
| 29.2.2.25 | PIC1AREG50 — Timer Input/Output Control Register 50..... | 2334 |
| 29.2.2.26 | PIC1AREG51 — Timer Input/Output Control Register 51..... | 2335 |
| 29.2.2.27 | POMONSEL — Port Output Monitor Select Register..... | 2336 |
| 29.2.2.28 | PIMONSEL — Port Input Monitor Select Register | 2337 |
| 29.2.2.29 | SELBSSER — Synchronous Clear Enable Register..... | 2338 |
| 29.2.3 | Function..... | 2339 |
| 29.2.3.1 | Simultaneous Start Trigger Function..... | 2339 |
| 29.2.3.2 | PWM Output Function with Dead Time | 2343 |
| 29.2.3.3 | High Accuracy Triangle Wave PWM Output Function with Dead Time..... | 2352 |
| 29.2.3.4 | Delay Pulse Output Function with Dead Time..... | 2365 |
| 29.2.3.5 | Trigger Pulse Interval Measurement Function | 2371 |
| 29.2.3.6 | Encoder Capture Trigger Select Function | 2379 |
| 29.2.3.7 | Two-Phase Encoder Control Function (Control Method 1) | 2386 |
| 29.2.3.8 | Two-Phase Encoder Control Function (Control Method 2) | 2392 |
| 29.2.3.9 | Three-Phase Pulse Input Control Function | 2398 |

| | | |
|-------------------|--|-------------|
| 29.2.3.10 | Three-Phase Encoder Control Function | 2407 |
| 29.2.3.11 | TAUD Input Select Function | 2411 |
| 29.2.3.12 | Hi-Z Control Function | 2413 |
| 29.2.3.13 | Timer Output Monitor Function (PWM-Diag) | 2416 |
| 29.2.3.14 | Timer Input Monitor Function | 2417 |
| 29.2.3.15 | TSG3 Synchronous Clear Function | 2418 |
| 29.3 | Peripheral Interconnection — 2 (PIC2B) | 2419 |
| 29.3.1 | Overview | 2419 |
| 29.3.1.1 | Functional Overview | 2419 |
| 29.3.2 | Registers | 2419 |
| 29.3.2.1 | List of Registers | 2419 |
| 29.3.2.2 | PIC2BADCGnTSELx — A/D Converter n Trigger Select Control Register x | 2420 |
| 29.3.2.3 | PIC2BADCGnEDGSEL — A/D Converter Trigger Edge Control Register | 2422 |
| 29.3.2.4 | PIC2BADTEN4nx — A/D Converter Trigger Output Select Control Register | 2423 |
| 29.3.2.5 | ADSYNCTRG — A/D Converter Synchronized Start Trigger Register | 2423 |
| 29.3.3 | Function | 2424 |
| 29.3.3.1 | ADCG Trigger Select Function | 2424 |
| Section 30 | A/D Converter (ADCG) | 2427 |
| 30.1 | Features of RH850/P1M-E ADCG | 2427 |
| 30.1.1 | Number of Units | 2427 |
| 30.1.2 | Register Base Address | 2427 |
| 30.1.3 | Clock Supply | 2428 |
| 30.1.4 | Interrupt Requests | 2428 |
| 30.1.5 | Reset Sources | 2428 |
| 30.1.6 | External Input/Output Signals | 2429 |
| 30.1.7 | Analog Channels and Track and Hold Function | 2430 |
| 30.1.8 | Virtual Channel | 2430 |
| 30.2 | Overview | 2431 |
| 30.2.1 | Functional Overview | 2431 |
| 30.2.2 | Block Diagram | 2433 |
| 30.2.3 | Scan Group (SG) | 2435 |
| 30.3 | Registers | 2436 |
| 30.3.1 | List of Registers | 2436 |
| 30.3.2 | ADCG0ADSYNSTCR — A/D Synchronization Start Control Register | 2439 |
| 30.3.3 | ADCG0ADTSYNSTCR — A/D Timer Synchronization Start Control Register | 2439 |
| 30.3.4 | ADCGnVCRj — Virtual Channel Register j | 2440 |
| 30.3.5 | ADCGnDRj — Data Register j | 2442 |
| 30.3.6 | ADCGnDIRj — Data Supplementary Information Register j | 2444 |
| 30.3.7 | ADCGnSMPCR — A/D Conversion Time Control Register | 2446 |
| 30.3.8 | ADCGnADHALTR — A/D Halt Register | 2447 |
| 30.3.9 | ADCGnADCR1 — A/D Control Register 1 | 2448 |
| 30.3.10 | ADCGnMPXCURCR — MPX Current Control Register | 2449 |
| 30.3.11 | ADCGnMPXCURR — MPX Current Register | 2450 |
| 30.3.12 | ADCGnMPXOWR — MPX Optional Wait Register | 2451 |
| 30.3.13 | ADCGnADCR2 — A/D Control Register 2 | 2452 |

| | | |
|----------|---|------|
| 30.3.14 | ADCGnADENDPz — A/D Conversion Monitor Virtual Channel Pointer z..... | 2453 |
| 30.3.15 | ADCGnTHSMPSTCR — T&H Sampling Start Control Register | 2454 |
| 30.3.16 | ADCGnTHSTPCR — T&H Stop Control Register | 2455 |
| 30.3.17 | ADCGnTHCR — T&H Control Register | 2456 |
| 30.3.18 | ADCGnTHAHLSTCR — T&H Group A Hold Start Control Register..... | 2457 |
| 30.3.19 | ADCGnTHBHLSTCR — T&H Group B Hold Start Control Register..... | 2458 |
| 30.3.20 | ADCGnTHACR — T&H Group A Control Register..... | 2459 |
| 30.3.21 | ADCGnTHBCR — T&H Group B Control Register..... | 2461 |
| 30.3.22 | ADCGnTHER — T&H Enable Register | 2463 |
| 30.3.23 | ADCGnTHGSR — T&H Group Select Register | 2464 |
| 30.3.24 | ADCGnSFTCR — Safety Control Register | 2465 |
| 30.3.25 | ADCGnTDCR — Pin Level Self-Diagnostic Control Register..... | 2466 |
| 30.3.26 | ADCGnODCR — Wiring Break Detection Control Register | 2467 |
| 30.3.27 | ADOPDIGn — Wiring-break Detection Pin Setting Register n (n = 0, 1) | 2469 |
| 30.3.28 | ADCGnULLMTBR0 to 2 — Upper-limit/Lower-limit Table Registers 0 to 2..... | 2470 |
| 30.3.29 | ADCGnECR — Error Clear Register | 2471 |
| 30.3.30 | ADCGnULER — Upper-limit/Lower-limit Error Register..... | 2472 |
| 30.3.31 | ADCGnOWER — Overwrite Error Register..... | 2473 |
| 30.3.32 | ADCGnPER — Parity Error Register..... | 2474 |
| 30.3.33 | ADCGnIDER — ID Error Register..... | 2475 |
| 30.3.34 | ADCGnSGSTCRx — Scan Group x Start Control Register | 2476 |
| 30.3.35 | ADCGnADTSTCRy — A/D Timer y Start Control Register | 2477 |
| 30.3.36 | ADCGnADTENDCRy — A/D Timer y End Control Register | 2478 |
| 30.3.37 | ADCGnSGCRx — Scan Group x Control Register | 2479 |
| 30.3.38 | ADCGnSGVCSPx — Scan Group x Start Virtual Channel Pointer | 2481 |
| 30.3.39 | ADCGnSGVCEPx — Scan Group x End Virtual Channel Pointer | 2482 |
| 30.3.40 | ADCGnSGMCYCRx — Scan Group x Multicycle Register | 2483 |
| 30.3.41 | ADCGnSGSRx — Scan Group x Status Register..... | 2484 |
| 30.3.42 | ADCGnADTIPRy — A/D Timer Initial Phase Register y..... | 2485 |
| 30.3.43 | ADCGnADTPRRy — A/D Timer Period Register y | 2486 |
| 30.3.44 | ADCGnULLMSRx — Scan Group x Upper-limit/Lower-limit Table Select Register | 2487 |
| 30.4 | Function..... | 2488 |
| 30.4.1 | A/D Conversion Function..... | 2488 |
| 30.4.1.1 | Example of Multicycle Scan Mode Operation..... | 2488 |
| 30.4.1.2 | Example of Continuous Scan Mode Operation | 2490 |
| 30.4.1.3 | Simultaneous Track and Hold Operation (THC Control)..... | 2491 |
| 30.4.1.4 | Example of Normal A/D Conversion Operation in Addition Mode..... | 2495 |
| 30.4.1.5 | Example of Using an External Analog Multiplexer (Port Output)..... | 2496 |
| 30.4.2 | Trigger Function | 2497 |
| 30.4.2.1 | Starting a Scan Group by Using a Hardware Trigger..... | 2497 |
| 30.4.2.2 | Starting a Scan Group by Using an A/D Timer Trigger | 2497 |
| 30.4.2.3 | Starting A/D Timer by Using a Hardware Trigger..... | 2498 |
| 30.4.3 | Suspend Function..... | 2499 |
| 30.4.3.1 | Example of Synchronous Suspend and Resume Operation | 2499 |
| 30.4.3.2 | Example of Asynchronous Suspend and Resume Operation | 2500 |
| 30.4.3.3 | Mixed Synchronous and Asynchronous Suspend Operations | 2500 |
| 30.4.4 | Self-Diagnostic Function..... | 2501 |

| | | |
|-------------------|--|-------------|
| 30.4.4.1 | Pin Level Self-Diagnostic Function..... | 2501 |
| 30.4.4.2 | A/D Conversion Circuit Self-Diagnostic Function | 2503 |
| 30.4.4.3 | Wiring-Break Detection | 2504 |
| 30.4.4.4 | Diagnosis of Wiring-Break Detection..... | 2505 |
| 30.4.4.5 | Pin-Level Self-Diagnosis of T&H Paths..... | 2506 |
| 30.4.5 | Interrupt Request Function | 2512 |
| 30.4.5.1 | Scan End Interrupt Request | 2512 |
| 30.4.5.2 | A/D Error Interrupt Request and A/D Parity Error Interrupt Request..... | 2513 |
| 30.4.6 | Combination of Wiring Break Detection and A/D Conversion for Same Physical Channel | 2515 |
| 30.5 | Operation | 2516 |
| 30.5.1 | Initial Settings | 2516 |
| 30.5.2 | Procedure for Starting A/D Conversion | 2517 |
| 30.5.3 | Procedure for Stopping A/D Conversion | 2518 |
| 30.6 | Definition of A/D Conversion Accuracy..... | 2519 |
| Section 31 | Functional Safety | 2520 |
| 31.1 | Overview..... | 2520 |
| 31.2 | ECC and EDC..... | 2521 |
| 31.2.1 | Overview..... | 2521 |
| 31.2.1.1 | ECC..... | 2521 |
| 31.2.1.2 | Address Parity | 2522 |
| 31.2.2 | Code Flash ECC and Address Parity | 2523 |
| 31.2.2.1 | Overview | 2523 |
| 31.2.2.2 | List of Registers..... | 2524 |
| 31.2.2.3 | Details of Registers | 2525 |
| 31.2.2.4 | Test Function..... | 2536 |
| 31.2.3 | Data Flash ECC..... | 2537 |
| 31.2.3.1 | Overview | 2537 |
| 31.2.3.2 | List of Registers..... | 2538 |
| 31.2.3.3 | Details of Registers | 2539 |
| 31.2.3.4 | Test Function..... | 2547 |
| 31.2.4 | Local RAM (CPU1) ECC | 2548 |
| 31.2.4.1 | Overview | 2548 |
| 31.2.4.2 | List of Registers..... | 2549 |
| 31.2.4.3 | Details of Registers | 2551 |
| 31.2.4.4 | Test Function..... | 2562 |
| 31.2.5 | Global RAM ECC..... | 2564 |
| 31.2.5.1 | Overview | 2564 |
| 31.2.5.2 | List of Registers..... | 2565 |
| 31.2.5.3 | Details of Registers | 2568 |
| 31.2.5.4 | Test Function..... | 2583 |
| 31.2.6 | Instruction Cache ECC and EDC | 2585 |
| 31.2.6.1 | Overview | 2585 |
| 31.2.6.2 | List of Registers..... | 2586 |
| 31.2.6.3 | Details of Registers | 2587 |
| 31.2.6.4 | Test Function..... | 2605 |
| 31.2.7 | DTS RAM ECC..... | 2606 |

| | | |
|-----------|---|------|
| 31.2.8 | ECC for Peripheral RAM (32 Bits) | 2606 |
| 31.2.8.1 | Overview | 2606 |
| 31.2.8.2 | List of Registers..... | 2608 |
| 31.2.8.3 | Details of Registers | 2610 |
| 31.2.8.4 | Test Function..... | 2619 |
| 31.2.9 | ECC on Data Transfer Path..... | 2620 |
| 31.2.9.1 | ECC protection on Bus..... | 2620 |
| 31.2.9.2 | Failure Detection Function for GRAM Access..... | 2620 |
| 31.2.9.3 | Fault Detection Function of Arbitration | 2621 |
| 31.2.9.4 | List of Registers..... | 2621 |
| 31.2.9.5 | Details of Registers | 2625 |
| 31.2.9.6 | Test Function..... | 2673 |
| 31.2.10 | ECC Function | 2675 |
| 31.2.10.1 | ECC code for RAM Modules exception instruction cache RAM (data) | 2675 |
| 31.2.10.2 | EDC Function for Instruction cache RAM (data) | 2677 |
| 31.2.10.3 | ECC Function for Code Flash | 2679 |
| 31.2.10.4 | ECC Function for Data Flash | 2682 |
| 31.3 | Lockstep | 2684 |
| 31.3.1 | List of Registers..... | 2684 |
| 31.3.2 | Details of Registers | 2685 |
| 31.3.2.1 | CMPTST0 — Comparator Test Register 0..... | 2685 |
| 31.3.2.2 | CMPTST1 — Comparator Test Register 1..... | 2686 |
| 31.3.2.3 | PDMA_COMP_CNTRL — PDMA Comparator Error Injection Control Register | 2687 |
| 31.3.3 | Usage Notes..... | 2688 |
| 31.4 | Memory Protection..... | 2689 |
| 31.4.1 | Overview..... | 2689 |
| 31.4.1.1 | Protection Mechanisms Inside the Processors | 2691 |
| 31.4.1.2 | Peripheral Register Protection with H-bus Guard (HBG) and P-bus Guard (PBG) | 2693 |
| 31.4.1.3 | Global RAM Protection..... | 2693 |
| 31.4.1.4 | Protectable Memory Regions..... | 2694 |
| 31.4.1.5 | Default MPU and Guard Configuration..... | 2694 |
| 31.4.2 | GRG | 2694 |
| 31.4.2.1 | List of Registers..... | 2695 |
| 31.4.2.2 | Details of Registers | 2697 |
| 31.4.3 | PBG..... | 2704 |
| 31.4.3.1 | List of Registers..... | 2718 |
| 31.4.3.2 | Details of Registers | 2731 |
| 31.4.4 | HBG..... | 2744 |
| 31.4.4.1 | Details of Registers | 2745 |
| 31.5 | Clock Monitor (CLMA) | 2752 |
| 31.5.1 | Features | 2752 |
| 31.5.1.1 | Clock Monitor Configurations | 2752 |
| 31.5.1.2 | Block Diagram | 2753 |
| 31.5.1.3 | External Input/Output Pins | 2753 |
| 31.5.2 | Functional Description..... | 2754 |
| 31.5.2.1 | Detection of Abnormal Clock Frequencies..... | 2754 |
| 31.5.2.2 | Resetting CLMA..... | 2756 |

| | | |
|-------------------|--|-------------|
| 31.5.2.3 | Self-Diagnosis | 2756 |
| 31.5.3 | Register | 2757 |
| 31.5.3.1 | Register Protection..... | 2757 |
| 31.5.3.2 | List of Registers..... | 2757 |
| 31.5.3.3 | Resetting the Registers | 2758 |
| 31.5.3.4 | CLMA _n CTL0 — CLMA _n Control Register 0 | 2759 |
| 31.5.3.5 | CLMA _n CMPL — CLMA _n Compare Register L..... | 2760 |
| 31.5.3.6 | CLMA _n CMPH — CLMA _n Compare Register H | 2760 |
| 31.5.3.7 | CLMA _n PCMD — CLMA _n Protection Command Register | 2761 |
| 31.5.3.8 | CLMA _n PS — CLMA _n Protection Status Register | 2761 |
| 31.5.3.9 | CLMATEST — CLMA Self-test Register | 2762 |
| 31.5.3.10 | CLMATESTS — CLMA Self-test Status Register | 2763 |
| 31.5.4 | Operation..... | 2764 |
| 31.5.4.1 | Write-Protected Registers | 2764 |
| 31.5.4.2 | Reset Procedure Using CLMATEST.RESCLM | 2766 |
| 31.6 | BIST..... | 2767 |
| 31.6.1 | List of Registers..... | 2767 |
| 31.6.2 | Details of Registers | 2768 |
| 31.6.3 | Confirming Completion of BIST with no Abnormalities Detected | 2783 |
| 31.6.4 | State Transition of Field BIST..... | 2784 |
| 31.6.5 | Note | 2784 |
| 31.7 | ECM..... | 2785 |
| 31.8 | CVM..... | 2785 |
| 31.9 | WDTA | 2785 |
| 31.10 | DCRA..... | 2785 |
| Section 32 | Error Control Module (ECM) | 2786 |
| 32.1 | Features of RH850/P1M-E ECM | 2786 |
| 32.1.1 | Units | 2786 |
| 32.1.2 | Register Base Address..... | 2786 |
| 32.1.3 | Clock Supply..... | 2786 |
| 32.1.4 | Interrupt Requests | 2787 |
| 32.1.5 | External Output Signals..... | 2787 |
| 32.2 | Overview..... | 2788 |
| 32.2.1 | Specification Overview | 2788 |
| 32.2.2 | Flash Option | 2789 |
| 32.3 | Block Diagram | 2790 |
| 32.3.1 | Error Input..... | 2790 |
| 32.3.2 | Operations for Error Output | 2794 |
| 32.3.3 | ERROROUT behavior at reset | 2794 |
| 32.3.3.1 | Enabling Dynamic Mode | 2794 |
| 32.3.3.2 | Disabling Dynamic Mode..... | 2795 |
| 32.3.4 | Error Status | 2795 |
| 32.3.5 | Writing to Protected Registers..... | 2795 |
| 32.3.5.1 | Protection Unlock Sequence | 2795 |
| 32.3.5.2 | Interrupt during Write Sequence | 2796 |

| | | |
|----------|---|------|
| 32.3.5.3 | Emulation Break during Write Sequence | 2797 |
| 32.3.6 | Timeout Function for Interrupt Processing | 2797 |
| 32.3.7 | Configuration lock..... | 2797 |
| 32.3.8 | Masking of error clear trigger registers..... | 2797 |
| 32.4 | Register Specification..... | 2799 |
| 32.4.1 | List of Registers..... | 2799 |
| 32.4.2 | ECMmESET — ECM Master/Checker Error Set Trigger Register (m = M/C)..... | 2804 |
| 32.4.3 | ECMmECLR — ECM Master/Checker Error Clear Trigger Register (m = M/C) | 2805 |
| 32.4.4 | ECMmESSTR0 — ECM Master/Checker Error Source Status Register 0 (m = M/C) | 2806 |
| 32.4.5 | ECMmESSTR1 — ECM Master/Checker Error Source Status Register 1 (m = M/C) | 2807 |
| 32.4.6 | ECMmESSTR2 — ECM Master/Checker Error Source Status Register 2 (m = M/C) | 2808 |
| 32.4.7 | ECMmPCMD0 — ECM Master/Checker Protection Command Register (m = M/C)..... | 2809 |
| 32.4.8 | ECMEPCFG — ECM Error Pulse Configuration Register..... | 2810 |
| 32.4.9 | ECMMICFG0 — ECM Maskable Interrupt Configuration Register 0..... | 2811 |
| 32.4.10 | ECMMICFG1 — ECM Maskable Interrupt Configuration Register 1..... | 2812 |
| 32.4.11 | ECMMICFG2 — ECM Maskable Interrupt Configuration Register 2..... | 2813 |
| 32.4.12 | ECMNMICFG0 — ECM Non-maskable Interrupt Configuration Register 0 | 2814 |
| 32.4.13 | ECMNMICFG1 — ECM Non-maskable Interrupt Configuration Register 1 | 2815 |
| 32.4.14 | ECMNMICFG2 — ECM Non-maskable Interrupt Configuration Register 2 | 2816 |
| 32.4.15 | ECMIRCFG0 — ECM Internal Reset Configuration Register 0..... | 2817 |
| 32.4.16 | ECMIRCFG1 — ECM Internal Reset Configuration Register 1..... | 2818 |
| 32.4.17 | ECMIRCFG2 — ECM Internal Reset Configuration Register 2..... | 2819 |
| 32.4.18 | ECMEMK0 — ECM Error Mask Register 0 | 2820 |
| 32.4.19 | ECMEMK1 — ECM Error Mask Register 1 | 2821 |
| 32.4.20 | ECMEMK2 — ECM Error Mask Register 2 | 2822 |
| 32.4.21 | ECMESSTC0 — ECM Error Source Status Clear Trigger Register 0 | 2823 |
| 32.4.22 | ECMESSTC1 — ECM Error Source Status Clear Trigger Register 1 | 2824 |
| 32.4.23 | ECMESSTC2 — ECM Error Source Status Clear Trigger Register 2..... | 2825 |
| 32.4.24 | ECMPCMD1 — ECM Protection Command Register | 2826 |
| 32.4.25 | ECMPS — ECM Protection Status Register | 2827 |
| 32.4.26 | ECMPE0 — ECM Pseudo Error Trigger Register 0 | 2828 |
| 32.4.27 | ECMPE1 — ECM Pseudo Error Trigger Register 1 | 2829 |
| 32.4.28 | ECMPE2 — ECM Pseudo Error Trigger Register 2 | 2830 |
| 32.4.29 | ECMDTMCTL — ECM Delay Timer Control Register..... | 2831 |
| 32.4.30 | ECMDTMR — ECM Delay Timer Register..... | 2832 |
| 32.4.31 | ECMDTMCMP — ECM Delay Timer Compare Register..... | 2833 |
| 32.4.32 | ECMDTMCFG0 — ECM Delay Timer Configuration Register 0 | 2834 |
| 32.4.33 | ECMDTMCFG1 — ECM Delay Timer Configuration Register 1 | 2835 |
| 32.4.34 | ECMDTMCFG2 — ECM Delay Timer Configuration Register 2 | 2836 |
| 32.4.35 | ECMDTMCFG3 — ECM Delay Timer Configuration Register 3 | 2837 |
| 32.4.36 | ECMDTMCFG4 — ECM Delay Timer Configuration Register 4 | 2838 |
| 32.4.37 | ECMDTMCFG5 — ECM Delay Timer Configuration Register 5 | 2839 |
| 32.4.38 | ECMEOCCFG — ECM Error Output Clear Invalidation Configuration Register | 2840 |
| 32.4.39 | ECMPEM — ECM Pseudo Error Mask Register | 2841 |

| | | |
|-------------------|--|-------------|
| 32.4.40 | ECMEPCTL — ECM Error Pulse Control Register | 2842 |
| Section 33 | Data CRC Function A (DCRA)..... | 2843 |
| 33.1 | Features of RH850/P1M-E DCRA | 2843 |
| 33.1.1 | Units and Channels | 2843 |
| 33.1.2 | Register Base Address..... | 2844 |
| 33.1.3 | Clock Supply..... | 2844 |
| 33.1.4 | Interrupt Requests | 2844 |
| 33.1.5 | Reset Sources | 2844 |
| 33.1.6 | External Input/Output Signals..... | 2844 |
| 33.2 | Overview | 2845 |
| 33.2.1 | Functional Overview | 2845 |
| 33.2.2 | Block Diagram | 2845 |
| 33.2.3 | Operation Circuit..... | 2846 |
| 33.3 | Registers..... | 2848 |
| 33.3.1 | List of Registers..... | 2848 |
| 33.3.2 | DCRAnCIN — CRCn Input Register | 2849 |
| 33.3.3 | DCRAnCOUT — CRCn Data Register..... | 2850 |
| 33.3.4 | DCRAnCTL — CRCn Control Register | 2851 |
| 33.4 | Operation | 2852 |
| Section 34 | On-Chip Debugging Unit (OCD) | 2853 |
| 34.1 | Debug Function | 2853 |
| 34.2 | Calibration Function..... | 2855 |
| 34.3 | Trace Control Function | 2855 |
| 34.4 | Peripheral Break Control | 2855 |
| 34.5 | Usage Notes for On-chip Debugging..... | 2856 |
| Section 35 | Flash Memory | 2857 |
| 35.1 | Features..... | 2857 |
| 35.2 | Structure of Memory | 2858 |
| 35.2.1 | Mapping of Code Flash Memory | 2858 |
| 35.2.2 | Mapping of Data Flash Memory | 2859 |
| 35.3 | Operating Modes Associated with Flash Memory | 2860 |
| 35.4 | Functions | 2861 |
| 35.4.1 | Functional Overview | 2861 |
| 35.5 | Serial Programming..... | 2866 |
| 35.5.1 | Environments for Programming..... | 2866 |
| 35.6 | Communication Modes | 2867 |
| 35.6.1 | Asynchronous Flash Programming Interface — 1-Wire UART | 2867 |
| 35.6.2 | Asynchronous Flash Programming Interface — 2-Wire UART | 2867 |
| 35.6.3 | Synchronous Flash Programming Interface CSI | 2867 |
| 35.6.4 | Selection of Communication Method..... | 2868 |

| | | |
|-------------------|---|-------------|
| 35.7 | Self-Programming..... | 2869 |
| 35.7.1 | Outline | 2869 |
| 35.7.2 | Background Operation..... | 2869 |
| 35.7.3 | Enabling Self-Programming..... | 2869 |
| 35.7.3.1 | FLMDCNT Register..... | 2870 |
| 35.7.4 | Write-Protected Registers | 2870 |
| 35.7.4.1 | Write Procedure to Write-Protected Registers | 2871 |
| 35.7.4.2 | Interrupt during Write Sequence | 2872 |
| 35.7.4.3 | Emulation Break during Write Sequence | 2873 |
| 35.7.4.4 | FLMDCNT Register Protection..... | 2873 |
| 35.8 | Reading Flash Memory..... | 2875 |
| 35.8.1 | Reading Code Flash Memory..... | 2875 |
| 35.8.2 | Reading Data Flash Memory..... | 2875 |
| 35.9 | Description of Registers..... | 2876 |
| 35.9.1 | Registers Related to Data Flash Memory..... | 2876 |
| 35.9.1.1 | EPRDCYCL — Data Flash Memory Read Cycle Setting Register | 2876 |
| 35.9.2 | Registers Related to Write and Erase Protect of Flash Memory | 2877 |
| 35.9.2.1 | FHVE15 — FHVE15 Control Register..... | 2877 |
| 35.9.2.2 | FHVE3 — FHVE3 Control Register..... | 2878 |
| 35.9.3 | Registers Related to Product Information..... | 2879 |
| 35.9.3.1 | PRDNAME _n — Product Name Storage Register (n = 1 to 4) | 2879 |
| 35.9.4 | Registers Related to Reset Vector | 2880 |
| 35.9.4.1 | GREG8 — Reset Vector 0 Register | 2880 |
| 35.10 | Option Bytes | 2881 |
| 35.10.1 | Option Byte Setting..... | 2881 |
| 35.10.1.1 | Setting of Option Byte 0 | 2882 |
| 35.10.1.2 | Setting of Option Byte 2 | 2883 |
| 35.10.2 | OPBT0 — Option Byte 0 | 2884 |
| 35.10.3 | OPBT2 — Option Byte 2 | 2886 |
| 35.11 | ECC Test Area..... | 2887 |
| 35.12 | Usage Note..... | 2888 |
| Section 36 | RAM | 2889 |
| 36.1 | Features..... | 2889 |
| 36.1.1 | List of On-chip RAM | 2889 |
| 36.2 | Overview..... | 2890 |
| 36.2.1 | Function overview..... | 2890 |
| 36.2.1.1 | Access..... | 2890 |
| 36.2.1.2 | Emulation RAM | 2890 |
| 36.2.1.3 | ECC..... | 2890 |
| 36.2.1.4 | RAM initialization..... | 2890 |
| 36.3 | Backup Register | 2891 |
| 36.3.1 | List of Registers..... | 2891 |
| 36.3.2 | BRAMDAT _n — Backup Register | 2891 |
| 36.4 | Emulation RAM..... | 2892 |

| | | |
|-------------------|--|-------------|
| 36.4.1 | Emulation RAM..... | 2892 |
| 36.4.2 | Code Flash Emulation Function Using the Emulation RAM | 2892 |
| 36.4.3 | EmulationRAM memory map..... | 2893 |
| 36.5 | Calibration Function..... | 2894 |
| 36.5.1 | Calibration Function Unit (CFU) | 2894 |
| 36.5.2 | CFU Register List | 2894 |
| 36.5.3 | TM_CC — Cache Clear Operation Register | 2895 |
| 36.5.4 | TM_ME — Tuning Memory Mapping Enable Register | 2896 |
| 36.5.5 | TM_MS — Tuning Memory Mapping Status Register | 2898 |
| 36.5.6 | TM_MA0 to 3 — Tuning Memory Mapping Address Registers 0 to 3 | 2899 |
| 36.6 | Usage Notes | 2900 |
| Section 37 | Electrical Specifications | 2901 |
| 37.1 | Overview..... | 2901 |
| 37.1.1 | General Measurement Conditions..... | 2901 |
| 37.1.1.1 | Common Conditions..... | 2901 |
| 37.1.1.2 | AC Characteristic Measurement Condition | 2902 |
| 37.2 | Absolute Maximum Ratings | 2903 |
| 37.3 | Supply Voltage Characteristics..... | 2904 |
| 37.4 | Oscillator Characteristics | 2904 |
| 37.5 | Characteristics of High-Speed Internal Oscillator Circuit..... | 2905 |
| 37.6 | PLL Characteristics..... | 2905 |
| 37.7 | Regulator Characteristics | 2905 |
| 37.8 | DC Characteristics..... | 2906 |
| 37.8.1 | Relationship between Power Name and Pin | 2906 |
| 37.8.2 | Buffer Characteristics | 2907 |
| 37.8.3 | Allowable Output Current | 2908 |
| 37.8.4 | Injection Current | 2908 |
| 37.8.5 | Input Capacitance..... | 2908 |
| 37.8.6 | Power Current Characteristics..... | 2909 |
| 37.9 | AC Characteristics..... | 2910 |
| 37.9.1 | Power UP/Down Timing | 2910 |
| 37.9.2 | Driving Ability..... | 2912 |
| 37.9.3 | Clock Output Timing..... | 2912 |
| 37.9.4 | Control Signal Timing | 2913 |
| 37.9.4.1 | Reset..... | 2913 |
| 37.9.4.2 | Interrupts/ADTRG..... | 2914 |
| 37.9.4.3 | Mode | 2915 |
| 37.9.5 | Timer Timing..... | 2915 |
| 37.9.6 | CSI Timing..... | 2916 |
| 37.9.6.1 | CSIG Timing..... | 2916 |
| 37.9.6.2 | CSIH Timing | 2917 |
| 37.9.6.3 | CSIG/CSIH Timing Charts..... | 2918 |
| 37.9.7 | SCI3 Timing..... | 2927 |
| 37.9.8 | RS-CANFD Timing | 2929 |

| | | |
|--------------------------------------|--|------|
| 37.9.9 | RLIN3 Timing..... | 2929 |
| 37.9.10 | FlexRay Timing..... | 2929 |
| 37.9.11 | PSI5 Timing..... | 2930 |
| 37.9.12 | RSENT Timing..... | 2930 |
| 37.10 | POC Characteristics | 2931 |
| 37.11 | Core Voltage Monitor Characteristics | 2931 |
| 37.12 | Temperature Sensor..... | 2931 |
| 37.13 | BIST Execution Time | 2932 |
| 37.14 | A/D Convertor Characteristics | 2932 |
| 37.15 | Code Flash Characteristics..... | 2934 |
| 37.16 | Data Flash Characteristics..... | 2935 |
| 37.17 | Debug Interface | 2936 |
| 37.17.1 | JTAG/Nexus Timing | 2936 |
| 37.17.2 | LDU 4-Wire Timing..... | 2937 |
| 37.18 | Thermal Characteristics..... | 2938 |
| Appendix A. List of Registers | | 2939 |
| Appendix B. Package Dimensions | | 3116 |

Section 1 Overview

RH850/P1M-E is a product series of Renesas Electronics' single-chip microcomputer RH850 family. This section describes the overview of RH850/P1M-E.

This product has Security installed. For the specification of Security function, please contact with agent/distributor.

1.1 Outline

This product is a 32-bit single-chip microcomputer with multiple CPUs, Code Flash, Data Flash, RAM modules, DMA controllers, many communication interfaces that are used in the automotive applications, A/D converters, timer units, etc.

The main features are as below.

(1) RH850 CPU

This product contains two RH850G3Ms: one operates as the master CPU for normal operation and another as a checker CPU that monitors the operation of the master CPU. The two CPUs operate in lock step mode.

The architecture of this dual-core CPU unit realizes the compact footprint of the programs by 2-byte basic instructions and high-level language compiler oriented instruction sets. This dual-core CPU unit has very quick interrupt response time so that it can support hard real-time applications.

(2) On-chip Code Flash and Data Flash

This chip has high-speed Code Flash from which the CPU can fetch the instructions and the constant data. Its capacity is up to 2MB. This Code Flash can be reprogrammed in the situation that the chip is mounted in the application systems.

This chip also has Data Flash with EEPROM emulation capability. Its capacity is up to 64KB.

(3) Rich peripheral functionality

The RH850/P1M-E incorporates standard communication controllers such as CAN, LIN, FlexRay, and serial communications in addition to timers and A/D converters best suited for the automobile chassis applications. In addition, the RH850/P1M-E incorporates SENT and PSI5 communication controllers that can be used for sensor interface.

(4) Functional safety support

This chip equips several dedicated functionalities including the Lock-Step Dual Core configuration for the CPU, the memory protection with ECC, the bus protection with ECC, the peripheral module protection, and voltage/clock monitors to support the functional safety standard (ASIL) required in the automotive applications.

1.2 Application Fields

- Automotive field (including chassis control system)

1.3 Specification Overview

Table 1.1 Features in Each Product (1/2)

| Product | | | RH850/P1M-E | | | | |
|---------------|-----------------------|-------------------------|--|---|---------------------|--|--|
| | | | 100-pin | | 144-pin | | |
| Package | | | LFQFP100 (14x14) | LFQFP144 (16x16) | LFQFP144 (20x20) | | |
| Product name | eVR | 1MB | R7F701382 | R7F701384 | R7F701386 | | |
| | | 2MB | R7F701376 | R7F701378 | R7F701380 | | |
| | DPS | 1MB | R7F701381 | R7F701383 | R7F701385 | | |
| | | 2MB | R7F701375 | R7F701377 | R7F701379 | | |
| CPU Subsystem | Frequency | | 160 MHz Main OSC = 16 MHz only | | | | |
| | Main Core | | 1 | | | | |
| | Lockstep | | Yes | | | | |
| | FPU | Double prec. | Yes | | | | |
| | MPU | | 16 ch | | | | |
| | Cache (Inst.) | | 16 KB 4 way | | | | |
| | RAM | Local RAM | | 128 KB | | | |
| | | Global RAM | | 64 KB | | | |
| | | Trace RAM | | 32 KB | | | |
| | | ERAM (Emulation RAM) | | 32 KB ^{*1} /8 KB ^{*2} | | | |
| | INTC | INTC1 | | 32 ch Redundant | | | |
| | | INTC2 | | 352 ch No Redundancy | | | |
| | DMA | DMAC | | 16 ch Redundant | | | |
| | | DTS | | 128 ch Redundant | | | |
| Flash Memory | Code Flash | | 2 MB/1 MB | | | | |
| | DATA Flash | | 64 KB ^{*1} /32 KB ^{*2} | | | | |
| Safety | ECM | | Yes | | | | |
| | CVM | | Yes | | | | |
| | BIST with status flag | | Yes | | | | |
| | ERROROUT | | Yes | | | | |
| | Clock monitor | | Yes | | | | |
| A/D Converter | ADC 12 bit | module | | 2 | | | |
| | | Temperature sensor | | Yes | | | |
| | | Analog input | ADCG0 | 9 ch | 12 ch | | |
| | | | ADCG1 | 10 ch | 12 ch | | |
| | | T & H | ADCG0 | 6 | | | |
| | | | ADCG1 | 4 | | | |

Table 1.1 Features in Each Product (2/2)

| Product | | | RH850/P1M-E | |
|-------------------------|--------------------------------------|----------|-----------------|---------|
| | | | 100-pin | 144-pin |
| Timer | TAUD (16 bit x 16 ch) | modules | 3 | |
| | TAUJ (32 bit x 4 ch) | modules | 3 | |
| | TSG3 | modules | 2 | |
| | TAPA (Hi-Z control) | modules | 4 | |
| | ENCA | modules | 2 | |
| | OSTM | modules | 5 | |
| | OSTM (Output) | modules | 2 | |
| | WDT | modules | 1 | |
| | TPBA (Timer Pattern Buffer) | modules | 2 | |
| | PIC (Peripheral interconnect) | | Yes | |
| | Clock out | modules | 2 | |
| Communication Interface | RS-CANFD | channels | 3 | |
| | FlexRay | channels | 2 | |
| | PSI5 | channels | 2 | |
| | RSENT | channels | 5 | 6 |
| | SCI 3 (Sync/Async USART) | channels | 3 | |
| | RLIN3 (LIN master/UART) | channels | 2 | |
| | CSIG (SPI) | channels | 1 | |
| | CSIH (SPI) | channels | 4 | |
| DataCRC | | | 4 | |
| Security (ICUS) | | | Yes | |
| Debug | Nexus-JTAG | | Yes | |
| | LDU | | Yes | |
| | AUD-RAM monitor | | No | |
| | Branch Trace (for Debug) (Output) | | No | |
| | Data Trace (for Debug) (Output) | | No | |
| | Internal Branch Trace (to Trace RAM) | | Yes | |
| | Internal Data Trace (to Trace RAM) | | Yes | |
| | Measurement (Memory Read) | | Yes | |
| Power Supply | Core | | 1.25 (DPS), eVR | |
| | I/O | | 3.3/5.0 | |
| | ADC | | 3.3/5.0 | |

Note 1. Only available in devices with 2-MB flash memory.

Note 2. Only available in devices with 1-MB flash memory.

1.4 Block Configuration

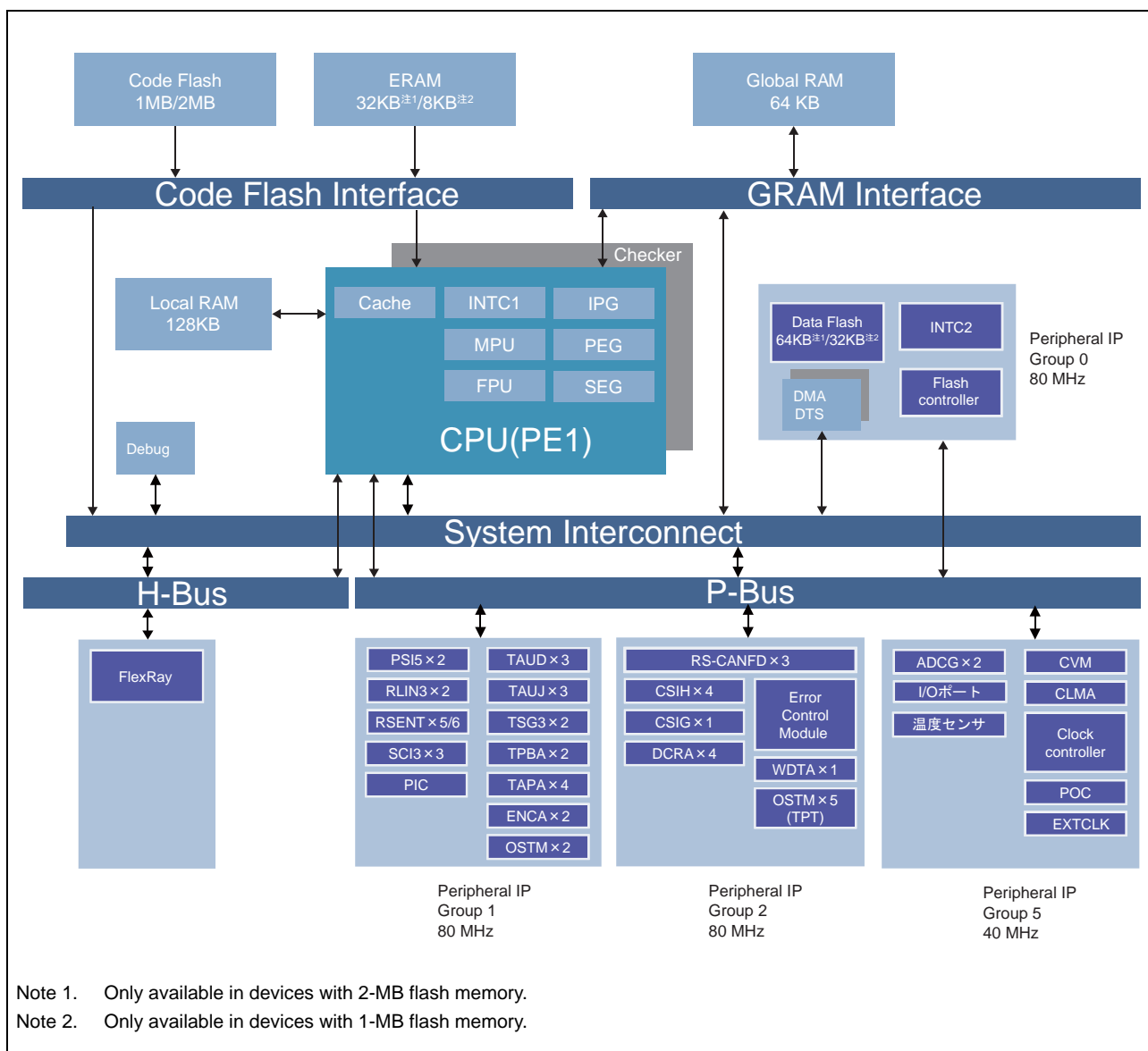


Figure 1.1 Internal Block Diagram

Section 2 Pin Functions

2.1 Pin Connection Diagrams

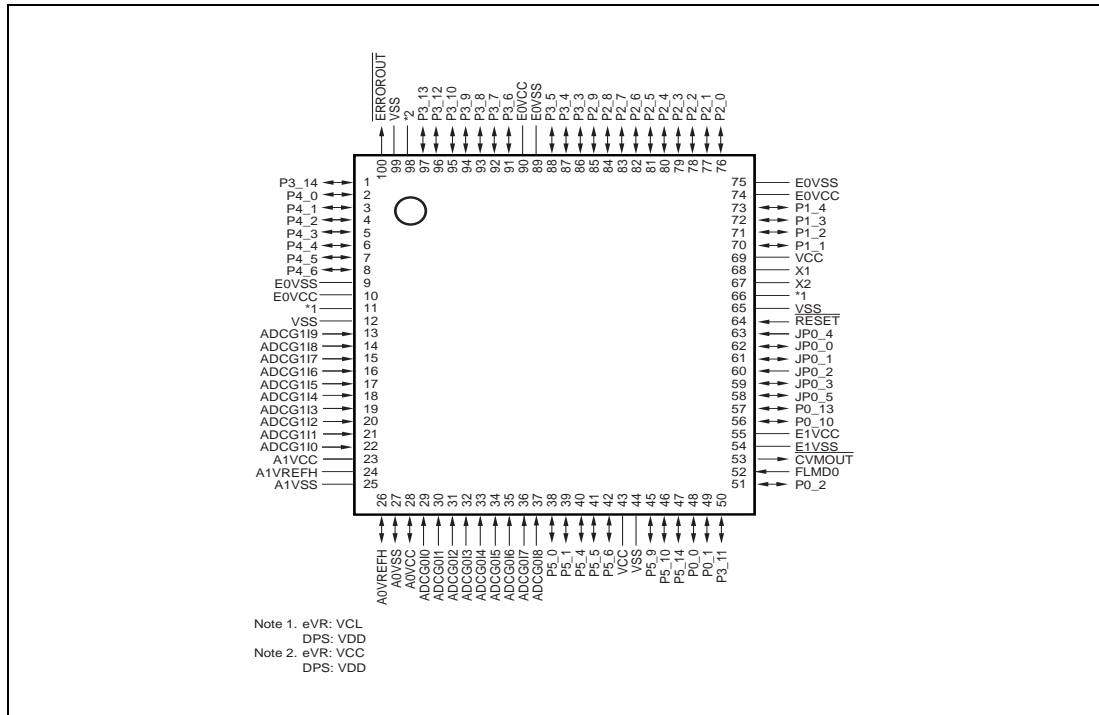


Figure 2.1 Pin Connection Diagram (100 pins)

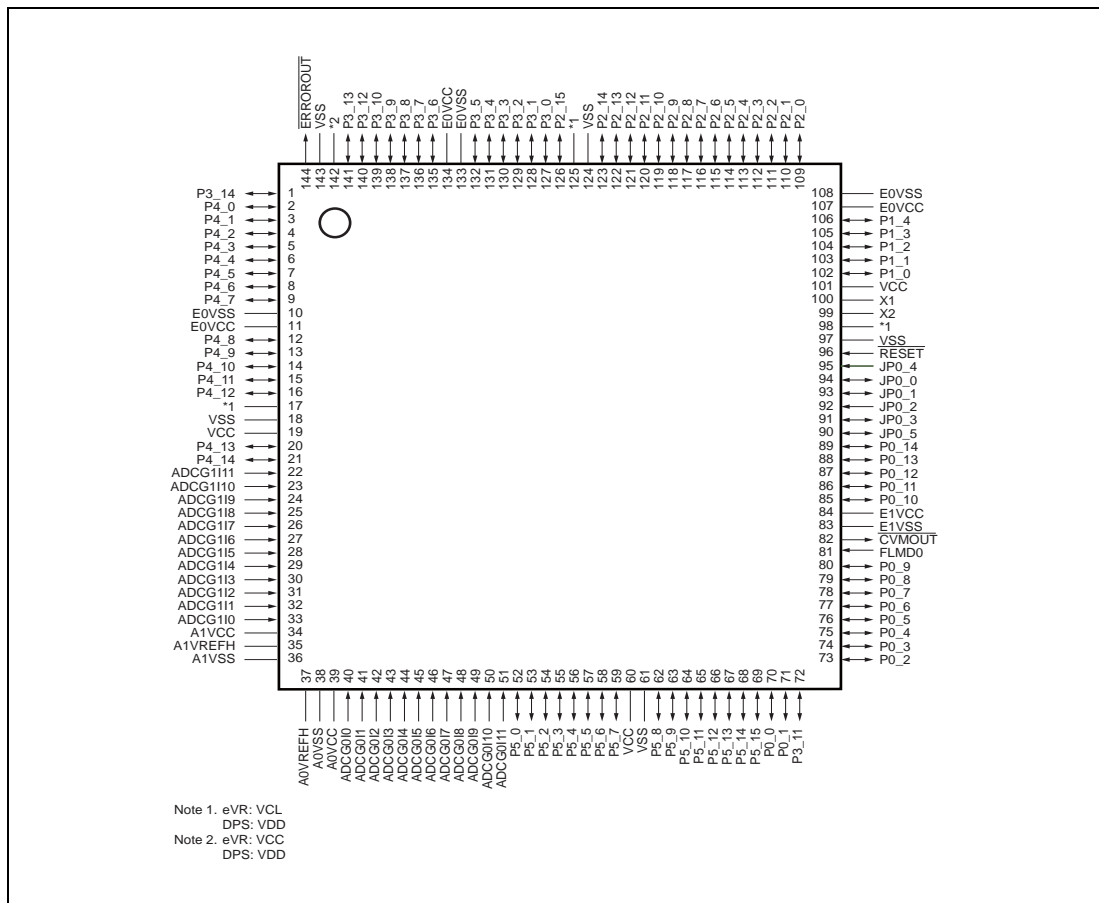


Figure 2.2 Pin Connection Diagram (144 pins)

Table 2.1 shows the pin assignment.

CAUTION

There are restrictions on the combinations of the CSIG, CSIH, SCI3, RLIN3, RSCAN, FlexRay, and RSENT pins that can be used. For details, see the Combinations of Pins and Ports table in each section.

Table 2.1 Pin Assignment (1/5)

| Pin Name | Pin Number | | | | Power Area |
|---|---------------|---------------|---------------|---------------|------------|
| | Pkg 100 (eVR) | Pkg 100 (DPS) | Pkg 144 (eVR) | Pkg 144 (DPS) | |
| P3_14 / CSIH1SI / TAPA0ESO / INTP7 / TPBA1O / EXTCLK0O / NMI / TSG31O7/FLMD1 | 1 | 1 | 1 | 1 | E0VCC |
| P4_0 / CSIH1SO / TAUJ0I0 / TAUJ0O0 / RLIN30RX / INTP3 / TSG30PTSIO / ENCA0E0 / FLXA0TXDA / ADCG1CNV0 | 2 | 2 | 2 | 2 | E0VCC |
| P4_1 / CSIH1SCI / CSIH1SCO / TAUJ0I1 / TAUJ0O1 / RLIN30TX / TSG30PTS1I / ENCA0E1 / FLXA0TXENA / ADCG1CNV1 | 3 | 3 | 3 | 3 | E0VCC |
| P4_2 / RSCAN0RX1 / INTP6 / CSIH1RYO / TAUJ0I2 / TAUJ0O2 / SCI30SCI / SCI30SCO / TSG30PTS2I / ENCA0EC / FLXA0RXDA / INTP11 / ADCG1CNV2 | 4 | 4 | 4 | 4 | E0VCC |
| P4_3 / CSIH2RY1 / RSCAN0TX1 / TAUJ0I3 / TAUJ0O3 / OSTM1O / TSG30CLKI / CSIH1CSS0 / FLXA0RXDB / INTP12 / ADCG1CNV3 / TAUD2I0 / TAUD2O0 | 5 | 5 | 5 | 5 | E0VCC |
| P4_4 / CSIH2SI / TAPA1ESO / INTP8 / CSIH2CSS7 / TSG31PTSIO / ENCA1E0 / CSIH1CSS1 / FLXA0STPWT / ADCG1CNV4 / TAUD2I1 / TAUD2O1 | 6 | 6 | 6 | 6 | E0VCC |
| P4_5 / CSIH2SO / SCI30RX / INTP0 / RSCAN0RX0 / INTP5 / EXTCLK1O / TSG31PTS1I / ENCA1E1 / CSIH1CSS2 / FLXA0TXDB / TAUD2I2 / TAUD2O2 | 7 | 7 | 7 | 7 | E0VCC |
| P4_6 / CSIH2SCI / CSIH2SCO / SCI30TX / RSCAN0TX0 / TSG31PTS2I / ENCA1EC / CSIH1CSS3 / FLXA0TXENB / TAUD2I3 / TAUD2O3 | 8 | 8 | 8 | 8 | E0VCC |
| P4_7 / RSCAN0RX1 / INTP6 / CSIH2RYO / TAUD1O0 / ENCA0TIN0 / CSIH2CSS0 / TSG31CLKI / CSIH1CSS4 / TAUD2I4 / TAUD2O4 | — | — | 9 | 9 | E0VCC |
| E0VSS | 9 | 9 | 10 | 10 | — |
| E0VCC | 10 | 10 | 11 | 11 | — |
| P4_8 / FLXA0RXDA / INTP11 / TAUD1O1 / ENCA0TIN1 / CSIH2CSS1 / CSIH1SSI / CSIH1CSS5 / TAUD2I5 / TAUD2O5 | — | — | 12 | 12 | E0VCC |
| P4_9 / FLXA0TXDA / TAUD1O2 / ENCA1TIN0 / CSIH2CSS2 / CSIH1RY1 / CSIH1CSS6 / TAUD2I6 / TAUD2O6 | — | — | 13 | 13 | E0VCC |
| P4_10 / FLXA0TXENA / TAUD1O3 / ENCA1TIN1 / CSIH2CSS3 / CSIH1RYO / CSIH1CSS7 / TAUD2I7 / TAUD2O7 | — | — | 14 | 14 | E0VCC |
| P4_11 / FLXA0RXDB / INTP12 / TAUD1O4 / CSIH2CSS4 / TAUJ1I0 / TAUJ1O0 / TAUD2I8 / TAUD2O8 | — | — | 15 | 15 | E0VCC |
| P4_12 / FLXA0TXDB / TAUD1O5 / CSIH2CSS5 / TAUJ1I1 / TAUJ1O1 / TAUD2I9 / TAUD2O9 | — | — | 16 | 16 | E0VCC |
| VCL | 11 | — | 17 | — | — |
| VDD | — | 11 | — | 17 | — |
| VSS | 12 | 12 | 18 | 18 | — |
| VCC | — | — | 19 | 19 | — |
| P4_13 / FLXA0TXENB / TAUD1O6 / CSIH2SSI / TAUJ1I2 / TAUJ1O2 / TAUD2I10 / TAUD2O10 | — | — | 20 | 20 | E0VCC |
| P4_14 / FLXA0STPWT / TAUD1O7 / CSIH2CSS6 / TAUJ1I3 / TAUJ1O3 / TAUD2I11 / TAUD2O11 | — | — | 21 | 21 | E0VCC |
| ADCG1I11 | — | — | 22 | 22 | A1VCC |
| ADCG1I10 | — | — | 23 | 23 | A1VCC |
| ADCG1I9 | 13 | 13 | 24 | 24 | A1VCC |

Table 2.1 Pin Assignment (2/5)

| Pin Name | Pin Number | | | | Power Area |
|--|---------------|---------------|---------------|---------------|------------|
| | Pkg 100 (eVR) | Pkg 100 (DPS) | Pkg 144 (eVR) | Pkg 144 (DPS) | |
| ADCG1I8 | 14 | 14 | 25 | 25 | A1VCC |
| ADCG1I7 | 15 | 15 | 26 | 26 | A1VCC |
| ADCG1I6 | 16 | 16 | 27 | 27 | A1VCC |
| ADCG1I5 | 17 | 17 | 28 | 28 | A1VCC |
| ADCG1I4 | 18 | 18 | 29 | 29 | A1VCC |
| ADCG1I3 | 19 | 19 | 30 | 30 | A1VCC |
| ADCG1I2 | 20 | 20 | 31 | 31 | A1VCC |
| ADCG1I1 | 21 | 21 | 32 | 32 | A1VCC |
| ADCG1I0 | 22 | 22 | 33 | 33 | A1VCC |
| A1VCC | 23 | 23 | 34 | 34 | — |
| A1VREFH | 24 | 24 | 35 | 35 | — |
| A1VSS | 25 | 25 | 36 | 36 | — |
| A0VREFH | 26 | 26 | 37 | 37 | — |
| A0VSS | 27 | 27 | 38 | 38 | — |
| A0VCC | 28 | 28 | 39 | 39 | — |
| ADCG0I0 | 29 | 29 | 40 | 40 | A0VCC |
| ADCG0I1 | 30 | 30 | 41 | 41 | A0VCC |
| ADCG0I2 | 31 | 31 | 42 | 42 | A0VCC |
| ADCG0I3 | 32 | 32 | 43 | 43 | A0VCC |
| ADCG0I4 | 33 | 33 | 44 | 44 | A0VCC |
| ADCG0I5 | 34 | 34 | 45 | 45 | A0VCC |
| ADCG0I6 | 35 | 35 | 46 | 46 | A0VCC |
| ADCG0I7 | 36 | 36 | 47 | 47 | A0VCC |
| ADCG0I8 | 37 | 37 | 48 | 48 | A0VCC |
| ADCG0I9 | — | — | 49 | 49 | A0VCC |
| ADCG0I10 | — | — | 50 | 50 | A0VCC |
| ADCG0I11 | — | — | 51 | 51 | A0VCC |
| P5_0 / CSIG0SI / TAUD0O0 / TAUD0I1 / TAUD0O1 / CSIH2CSS1 / SCI30RX / INTP0 | 38 | 38 | 52 | 52 | E1VCC |
| P5_1 / CSIG0SO / TAUD0O2 / TAUD0I3 / ADCG0CNV0 / TAUD0O3 / TAUD0I2 / CSIH2CSS2 / TAUD0I11 / SCI30TX | 39 | 39 | 53 | 53 | E1VCC |
| P5_2 / CSIG0SCI / CSIG0SCO / TAUD0O2 / TAUD0I14 / TAUD0O3 / TAUD0I15 / CSIH2CSS0 / SCI30SCI / SCI30SCO | — | — | 54 | 54 | E1VCC |
| P5_3 / TAUD0I7 / TAUD0I6 | — | — | 55 | 55 | E1VCC |
| P5_4 / CSIG0SCI / CSIG0SCO / TAUD0O4 / TAUD0I5 / ADCG0CNV1 / TAUD0O5 / TAUD0I4 / CSIH2CSS3 / SCI30SCI / SCI30SCO | 40 | 40 | 56 | 56 | E1VCC |
| P5_5 / SENT0RX / SENT0SPCO / TAUD0O6 / TAUD0I7 / ADCG0CNV2 / TAUD0O7 / TAUD0I6 / CSIH2CSS4 / SCI31RX / INTP1 / RSCAN0TX2 | 41 | 41 | 57 | 57 | E1VCC |
| P5_6 / INTP10 / RSCAN0RX2 / SENT0SPCO / TAUD0O8 / TAUD0I9 / ADCG0CNV3 / TAUD0O9 / TAUD0I8 / CSIH2CSS5 / SCI31TX | 42 | 42 | 58 | 58 | E1VCC |
| P5_7 / RSCAN0TX2 / TAUD0O10 / TAUD0I11 / ADCG0CNV4 / TAUD0O11 / TAUD0I10 / CSIH2CSS6 / SCI31SCI / SCI31SCO | — | — | 59 | 59 | E1VCC |
| VCC | 43 | 43 | 60 | 60 | — |
| VSS | 44 | 44 | 61 | 61 | — |
| P5_8 / SENT1RX / SENT1SPCO / TAUD0O12 / TAUD0I13 / TAUD0O13 / TAUD0I12 / CSIH2CSS7 / SCI32RX / INTP2 | — | — | 62 | 62 | E1VCC |

Table 2.1 Pin Assignment (3/5)

| Pin Name | Pin Number | | | | Power Area |
|---|---------------|---------------|---------------|---------------|------------|
| | Pkg 100 (eVR) | Pkg 100 (DPS) | Pkg 144 (eVR) | Pkg 144 (DPS) | |
| P5_9 / TAUD0I14 / SENT1SPCO / TAUD0O14 / TAUD0I15 / TAUD0O15 / PSI51DIN / CSH2SI / SCI32TX | 45 | 45 | 63 | 63 | E1VCC |
| P5_10 / RLIN30RX / INTP3 / TAUD0O12 / SENT1RX / SENT1SPCO / ADCGTRG0 / ADCG0CNV0 / TAUD0I12 / PSI51DOUT / SCI32SCI / SCI32SCO | 46 | 46 | 64 | 64 | E1VCC |
| P5_11 / RLIN30TX / SENT3RX / SENT3SPCO / TAUD1O10 / CSH2SSI | — | — | 65 | 65 | E1VCC |
| P5_12 / RLIN31RX / INTP4 / SENT2RX / SENT2SPCO / TAUD1O11 / CSH2RYI | — | — | 66 | 66 | E1VCC |
| P5_13 / RLIN31TX / SENT2SPCO / TAUD1O12 / TAUD0I13 / TAUD0I12 / CSH2RYO | — | — | 67 | 67 | E1VCC |
| P5_14 / TAUJ0I0 / TAUJ0O0 / SENT3RX / SENT3SPCO / TAUD1O13 / ADCGTRG1 / ADCG0CNV1 / PSI50DIN / CSH2SO / RLIN30RX / INTP3 | 47 | 47 | 68 | 68 | E1VCC |
| P5_15 / RLIN31RX / INTP4 / SENT3SPCO / CSH2SCI / CSH2SCO / RLIN30TX | — | — | 69 | 69 | E1VCC |
| P0_0 / TAUJ0I1 / TAUJ0O1 / TAUD0I13 / SENT3SPCO / TAUD1O14 / ADCG0CNV2 / TSG31CLKI / PSI50DOUT / RLIN30TX | 48 | 48 | 70 | 70 | E1VCC |
| P0_1 / TAUJ0I2 / TAUJ0O2 / SENT4RX / SENT4SPCO / TAUD1O15 / ADCG0CNV3 / TAUD0O10 / RLIN31RX / INTP4 | 49 | 49 | 71 | 71 | E1VCC |
| P3_11 / TAUJ0I3 / TAUJ0O3 / SENT4SPCO / ADCG0CNV4 / ERROROUT_C / RLIN31TX | 50 | 50 | 72 | 72 | E1VCC |
| P0_2 / SCI30RX / INTP0 / TAPA1ESO / INTP8 / TAUD2I0 / ADCG0CNV0 / TAUD2O0 / SCI32RX / INTP2 / TSG31O0 / SENT5RX / SENT5SPCO | 51 | 51 | 73 | 73 | E1VCC |
| P0_3 / SCI30TX / TAUD1O8 / TAUD2I1 / ADCG0CNV1 / TAUD2O1 / TSG31O1 / SENT5SPCO | — | — | 74 | 74 | E1VCC |
| P0_4 / SCI30SCI / SCI30SCO / TAUD1O9 / TAUD2I2 / ADCG0CNV2 / TAUD2O2 / TSG31O2 / SENT4SPCO | — | — | 75 | 75 | E1VCC |
| P0_5 / TAUJ1I0 / TAUJ1O0 / TAUD1O10 / TAUD2I3 / ADCG0CNV3 / TAUD2O3 / TSG31O3 / SENT3SPCO | — | — | 76 | 76 | E1VCC |
| P0_6 / TAUJ1I1 / TAUJ1O1 / TAUD1O11 / TAUD2I4 / TAUD2O4 / TSG31O4 / SENT2SPCO | — | — | 77 | 77 | E1VCC |
| P0_7 / TAUJ1I2 / TAUJ1O2 / TAUD1O12 / TAUD2I5 / ADCGTRG0 / TAUD2O5 / TSG31O5 / SENT1SPCO | — | — | 78 | 78 | E1VCC |
| P0_8 / TAUJ1I3 / TAUJ1O3 / TAUD1O13 / TAUD2I6 / ADCGTRG1 / TAUD2O6 / TSG31O6 / SENT0SPCO | — | — | 79 | 79 | E1VCC |
| P0_9 / ADCG1CNV1 / TAUD1O14 / TAUD2I7 / ADCG0CNV4 / TAUD2O7 / TSG31O7 | — | — | 80 | 80 | E1VCC |
| FLMD0 | 52 | 52 | 81 | 81 | E1VCC |
| CVMOUT | 53 | 53 | 82 | 82 | E1VCC |
| E1VSS | 54 | 54 | 83 | 83 | — |
| E1VCC | 55 | 55 | 84 | 84 | — |
| P0_10 / RESETOUT / ADCG1CNV0 / EVTO | 56 | 56 | 85 | 85 | E1VCC |
| P0_11 / ADCG1CNV2 / TSG31PTSIO / ENCA1E0 | — | — | 86 | 86 | E1VCC |
| P0_12 / ADCG1CNV3 / TSG31PTSII / ENCA1E1 | — | — | 87 | 87 | E1VCC |
| P0_13 / OSTM00 / TAUD1O15 / TAUD2I7 / INTP9 / TAUD2O7 / EVTI / TSG31PTSII / ENCA1EC / SCI31SCI / SCI31SCO | 57 | 57 | 88 | 88 | E1VCC |
| P0_14 / ADCG1CNV4 | — | — | 89 | 89 | E1VCC |
| JP0_5 / DCUTRDY / LPDCLKOUT | 58 | 58 | 90 | 90 | E1VCC |
| JP0_3 / DCUTMS | 59 | 59 | 91 | 91 | E1VCC |
| JP0_2 / FLSCI3SCKI (FPCK) / DCUTCK / LPDCLK | 60 | 60 | 92 | 92 | E1VCC |
| JP0_1 / FLSCI3TXD (FPDT) / DCUTDO / LPDO | 61 | 61 | 93 | 93 | E1VCC |
| JP0_0 / FLSCI3RXD (FPDR) / FLSCI3TXD (FPDT) / DCUTDI / LPDI | 62 | 62 | 94 | 94 | E1VCC |

Table 2.1 Pin Assignment (4/5)

| Pin Name | Pin Number | | | | Power Area |
|--|---------------|---------------|---------------|---------------|------------|
| | Pkg 100 (eVR) | Pkg 100 (DPS) | Pkg 144 (eVR) | Pkg 144 (DPS) | |
| JP0_4 / $\overline{\text{DCUTRST}}$ | 63 | 63 | 95 | 95 | VCC |
| RESET | 64 | 64 | 96 | 96 | VCC |
| VSS | 65 | 65 | 97 | 97 | — |
| VDD | — | 66 | — | 98 | — |
| VCL | 66 | — | 98 | — | — |
| X2 | 67 | 67 | 99 | 99 | VCC |
| X1 | 68 | 68 | 100 | 100 | VCC |
| VCC | 69 | 69 | 101 | 101 | — |
| P1_0 / RLIN30TX / CSIH3RYO | — | — | 102 | 102 | E0VCC |
| P1_1 / TAUJ2I0 / TAUJ2O0 / TAUJ1I0 / TAUJ1O0 / TAUD2I12 / TAUD2O12 / RLIN30RX / INTP3 / CSIH2CSS0 / TAUD0I6 / CSIH3RYI | 70 | 70 | 103 | 103 | E0VCC |
| P1_2 / TAUJ2I1 / TAUJ2O1 / TAUJ1I1 / TAUJ1O1 / TAUD2I13 / TAUD2O13 / CSIH2SI / TAUD0I8 / CSIH3SI | 71 | 71 | 104 | 104 | E0VCC |
| P1_3 / TAUJ2I2 / TAUJ2O2 / TAUJ1I2 / TAUJ1O2 / TAUD2I14 / TAUD2O14 / CSIH2SO / TAUD0I10 / CSIH3SO | 72 | 72 | 105 | 105 | E0VCC |
| P1_4 / TAUJ2I3 / TAUJ2O3 / TAUJ1I3 / TAUJ1O3 / TAUD2I15 / TAUD2O15 / CSIH2SCI / CSIH2SCO / CSIH3SCI / CSIH3SCO | 73 | 73 | 106 | 106 | E0VCC |
| E0VCC | 74 | 74 | 107 | 107 | — |
| E0VSS | 75 | 75 | 108 | 108 | — |
| P2_0 / RSCAN0RX0 / INTP5 / CSIH2SI / TAUD2I11 / TAUD2O11 / CSIH3CSS6 | 76 | 76 | 109 | 109 | E0VCC |
| P2_1 / RSCAN0TX0 / CSIH2SO / TAUD2I12 / TAUD2O12 / CSIH3CSS7 / TSG30O7 | 77 | 77 | 110 | 110 | E0VCC |
| P2_2 / RSCAN0RX1 / INTP6 / CSIH2SCI / CSIH2SCO / TAUD2I13 / TAUD2O13 / TPBA0O / TSG30O0 / CSIH0CSS5 | 78 | 78 | 111 | 111 | E0VCC |
| P2_3 / RSCAN0TX1 / CSIH2RYI / CSIH2CSS0 / TAUD2I14 / TAUD2O14 / TPBA1O / TSG30O1 | 79 | 79 | 112 | 112 | E0VCC |
| P2_4 / CSIH2RYO / CSIH0CSS2 / RLIN31TX / TAUJ1I3 / TAUJ1O3 / CSIH0SI / TAUD2I11 / TAUD2O11 | 80 | 80 | 113 | 113 | E0VCC |
| P2_5 / SCI30RX / INTP0 / CSIH0CSS3 / RLIN31RX / INTP4 / CSIH3SO / TSG30O2 / CSIH0SO | 81 | 81 | 114 | 114 | E0VCC |
| P2_6 / SCI30TX / OSTM1O / CSIH0SCI / CSIH0SCO / CSIH3SI / CSIH0CSS4 / TSG30O3 / TAUD1I0 / TAUD1O0 | 82 | 82 | 115 | 115 | E0VCC |
| P2_7 / SCI30SCI / SCI30SCO / CSIH0CSS5 / CSIH1SI / CSIH3SCI / CSIH3SCO / TSG30O4 / TAUD1I1 / TAUD1O1 | 83 | 83 | 116 | 116 | E0VCC |
| P2_8 / SCI31RX / INTP1 / CSIH3RYO / CSIH0CSS6 / CSIH1SO / CSIH3RYI / CSIH3CSS0 / TSG30O5 / TAUD1I2 / TAUD1O2 | 84 | 84 | 117 | 117 | E0VCC |
| P2_9 / SCI31TX / CSIH0CSS7 / CSIH1SCI / CSIH1SCO / CSIH3CSS1 / TSG30O6 / TAUD1I3 / TAUD1O3 | 85 | 85 | 118 | 118 | E0VCC |
| P2_10 / SENT5RX / SENT5SPCO / TAUD2I10 / TAUD2O10 / CSIH0RYI / CSIH0CSS6 / OSTM1O | — | — | 119 | 119 | E0VCC |
| P2_11 / TAUD1I0 / SENT5SPCO / TAUD1O0 / TAUD1I1 / TAUD1O1 / CSIH0CSS5 / CSIH1RYI / CSIH0RYO | — | — | 120 | 120 | E0VCC |
| P2_12 / TAUD1I2 / TAUD1O2 / TAUD1I3 / TAUD1O3 / CSIH0CSS4 / CSIH0SO | — | — | 121 | 121 | E0VCC |
| P2_13 / TAUD1I4 / TAUD1O4 / TAUD1I5 / TAUD1O5 / CSIH0CSS3 / CSIH1RYO | — | — | 122 | 122 | E0VCC |
| P2_14 / TAUD1I6 / TAUD1O6 / TAUD1I7 / TAUD1O7 / CSIH0CSS2 | — | — | 123 | 123 | E0VCC |
| VSS | — | — | 124 | 124 | — |
| VCL | — | — | 125 | — | — |
| VDD | — | — | — | 125 | — |
| P2_15 / TAUD1I8 / TAUD1O8 / TAUD1I9 / TAUD1O9 / CSIH0RYI | — | — | 126 | 126 | E0VCC |

Table 2.1 Pin Assignment (5/5)

| Pin Name | Pin Number | | | | Power Area |
|--|---------------|---------------|---------------|---------------|------------|
| | Pkg 100 (eVR) | Pkg 100 (DPS) | Pkg 144 (eVR) | Pkg 144 (DPS) | |
| P3_0 / TAUD1I10 / TAUD1O10 / TAUD1I11 / TAUD1O11 / CSIH0SO / CSIG0SI | — | — | 127 | 127 | E0VCC |
| P3_1 / TAUD1I12 / TAUD1O12 / TAUD1I13 / TAUD1O13 / CSIH0SCI / CSIH0SCO / CSIG0SO | — | — | 128 | 128 | E0VCC |
| P3_2 / TAUD1I14 / TAUD1O14 / TAUD1I15 / TAUD1O15 / CSIH0SI / CSIH0CSS7 / CSIG0SCI / CSIG0SCO | — | — | 129 | 129 | E0VCC |
| P3_3 / CSIH0RYI / SCI32TX / TAUD1I0 / CSIH2CSS1 / CSIH3SSI / CSIH3CSS4 / TAUD1O0 / TAUD1O1 | 86 | 86 | 130 | 130 | E0VCC |
| P3_4 / SCI32RX / INTP2 / TAUD1I2 / CSIH2CSS2 / RLIN30RX / INTP3 / CSIG0RYO*1 / CSIH3CSS3 / TAUD1O2 / TAUD1O3 | 87 | 87 | 131 | 131 | E0VCC |
| P3_5 / SCI31SCI / SCI31SCO / TAUD0I0 / CSIH2CSS3 / CSIG0RYI *1 / RLIN30TX / CSIH3CSS2 / TAUD0O0 / TAUD0O1 | 88 | 88 | 132 | 132 | E0VCC |
| E0VSS | 89 | 89 | 133 | 133 | — |
| E0VCC | 90 | 90 | 134 | 134 | — |
| P3_6 / CSIH0SI / TSG31O0 / TAUD0O2 / CSIH3RYI / CSIH2CSS4 / TAUD0O3 | 91 | 91 | 135 | 135 | E0VCC |
| P3_7 / CSIH0SO / TSG31O1 / RSCAN0RX0 / INTP5 / SCI30RX / INTP0 / CSIH2RYI / CSIH2CSS5 / TAUD0O5 | 92 | 92 | 136 | 136 | E0VCC |
| P3_8 / CSIH0SCI / CSIH0SCO / TSG31O2 / RSCAN0TX0 / TPBA1O / CSIH0SSI / CSIH2CSS6 / TAUD0O7 | 93 | 93 | 137 | 137 | E0VCC |
| P3_9 / SCI32SCI / SCI32SCO / TSG31O3 / CSIH3CSS5 / CSIH0RYI / CSIH2CSS7 / TAUD0O9 | 94 | 94 | 138 | 138 | E0VCC |
| P3_10 / TSG31O4 / TAUD1I6 / TAUD0O11 | 95 | 95 | 139 | 139 | E0VCC |
| P3_12 / CSIH1RYO / TSG31O5 / RSCAN0RX1 / INTP6 / SCI30TX / TAUD1I8 / CSIH0CSS0 / TAUD0O13 | 96 | 96 | 140 | 140 | E0VCC |
| P3_13 / CSIH1RYI / TSG31O6 / RSCAN0TX1 / SCI30SCI / SCI30SCO / TAUD1I10 / CSIH0CSS1 / TAUD0O15 | 97 | 97 | 141 | 141 | E0VCC |
| VDD | — | 98 | — | 142 | — |
| VCC | 98 | — | 142 | — | — |
| VSS | 99 | 99 | 143 | 143 | — |
| ERROROUT | 100 | 100 | 144 | 144 | E0VCC |

Note 1. Only in 144-pin products.

2.2 Pin Expansion

2.2.1 Overview

This subsection describes pin functions, external pin lists, and external pin states at a reset and in each operating status.

2.2.2 List of Pin Functions

Table 2.2 lists functions of each pin. “Port number” in the table indicates that the function is supported, while “—” indicates that the function is not supported. Do not use the pin function that is not supported.

Table 2.2 Pin Function (1/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------|----|---------------------------------------|--|--|--|--|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| NMI | NMI | I | External non-maskable interrupt input | P3_14 | P3_14 | P3_14 | P3_14 |
| INTP | INTP0 | I | External interrupt input 0 | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 |
| | INTP1 | I | External interrupt input 1 | P2_8/ P5_5 | P2_8/ P5_5 | P2_8/ P5_5 | P2_8/ P5_5 |
| | INTP2 | I | External interrupt input 2 | P0_2/ P3_4 | P0_2/ P3_4 | P0_2/ P3_4/ P5_8 | P0_2/ P3_4/ P5_8 |
| | INTP3 | I | External interrupt input 3 | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 |
| | INTP4 | I | External interrupt input 4 | P2_5/ P0_1 | P2_5/ P0_1 | P2_5/ P0_1/ P5_15/ P5_12 | P2_5/ P0_1/ P5_15/ P5_12 |
| | INTP5 | I | External interrupt input 5 | P3_7/ P2_0/ P4_5 | P3_7/ P2_0/ P4_5 | P3_7/ P2_0/ P4_5 | P3_7/ P2_0/ P4_5 |
| | INTP6 | I | External interrupt input 6 | P3_12/ P2_2/ P4_2 | P3_12/ P2_2/ P4_2 | P3_12/ P2_2/ P4_7/ P4_2 | P3_12/ P2_2/ P4_7/ P4_2 |
| | INTP7 | I | External interrupt input 7 | P3_14 | P3_14 | P3_14 | P3_14 |
| | INTP8 | I | External interrupt input 8 | P0_2/ P4_4 | P0_2/ P4_4 | P0_2/ P4_4 | P0_2/ P4_4 |
| | INTP9 | I | External interrupt input 9 | P0_13 | P0_13 | P0_13 | P0_13 |
| | INTP10 | I | External interrupt input 10 | P5_6 | P5_6 | P5_6 | P5_6 |
| | INTP11 | I | External interrupt input 11 | P4_2 | P4_2 | P4_8/ P4_2 | P4_8/ P4_2 |
| | INTP12 | I | External interrupt input 12 | P4_3 | P4_3 | P4_11/ P4_3 | P4_11/ P4_3 |
| TAUD0 | TAUD0I0 | I | TAUD0 channel input 0 | P3_5 | P3_5 | P3_5 | P3_5 |
| | TAUD0I1 | I | TAUD0 channel input 1 | P5_0 | P5_0 | P5_0 | P5_0 |
| | TAUD0I2 | I | TAUD0 channel input 2 | P5_1 | P5_1 | P5_1 | P5_1 |
| | TAUD0I3 | I | TAUD0 channel input 3 | P5_1 | P5_1 | P5_1 | P5_1 |
| | TAUD0I4 | I | TAUD0 channel input 4 | P5_4 | P5_4 | P5_4 | P5_4 |
| | TAUD0I5 | I | TAUD0 channel input 5 | P5_4 | P5_4 | P5_4 | P5_4 |

Table 2.2 Pin Function (2/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------|----|-------------------------|------------------|----------------|--------------------------|--------------------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| TAUD0 | TAUD0I6 | I | TAUD0 channel input 6 | P1_1/ P5_5 | P1_1/ P5_5 | P1_1/ P5_5/ P5_3 | P1_1/ P5_5/ P5_3 |
| | TAUD0I7 | I | TAUD0 channel input 7 | P5_5 | P5_5 | P5_5/ P5_3 | P5_5/ P5_3 |
| | TAUD0I8 | I | TAUD0 channel input 8 | P1_2/ P5_6 | P1_2/ P5_6 | P1_2/ P5_6 | P1_2/ P5_6 |
| | TAUD0I9 | I | TAUD0 channel input 9 | P5_6 | P5_6 | P5_6 | P5_6 |
| | TAUD0I10 | I | TAUD0 channel input 10 | P1_3 | P1_3 | P1_3/ P5_7 | P1_3/ P5_7 |
| | TAUD0I11 | I | TAUD0 channel input 11 | P5_1 | P5_1 | P5_1/ P5_7 | P5_1/ P5_7 |
| | TAUD0I12 | I | TAUD0 channel input 12 | P5_10 | P5_10 | P5_13/ P5_10/ P5_8 | P5_13/ P5_10/ P5_8 |
| | TAUD0I13 | I | TAUD0 channel input 13 | P0_0 | P0_0 | P5_13/ P5_8/ P0_0 | P5_13/ P5_8/ P0_0 |
| | TAUD0I14 | I | TAUD0 channel input 14 | P5_9 | P5_9 | P5_9/ P5_2 | P5_9/ P5_2 |
| | TAUD0I15 | I | TAUD0 channel input 15 | P5_9 | P5_9 | P5_9/ P5_2 | P5_9/ P5_2 |
| | TAUD0O0 | O | TAUD0 channel output 0 | P3_5/ P5_0 | P3_5/ P5_0 | P3_5/ P5_0 | P3_5/ P5_0 |
| | TAUD0O1 | O | TAUD0 channel output 1 | P3_5/ P5_0 | P3_5/ P5_0 | P3_5/ P5_0 | P3_5/ P5_0 |
| | TAUD0O2 | O | TAUD0 channel output 2 | P3_6/ P5_1 | P3_6/ P5_1 | P3_6/ P5_2/ P5_1 | P3_6/ P5_2/ P5_1 |
| | TAUD0O3 | O | TAUD0 channel output 3 | P3_6/ P5_1 | P3_6/ P5_1 | P3_6/ P5_2/ P5_1 | P3_6/ P5_2/ P5_1 |
| | TAUD0O4 | O | TAUD0 channel output 4 | P5_4 | P5_4 | P5_4 | P5_4 |
| | TAUD0O5 | O | TAUD0 channel output 5 | P3_7/ P5_4 | P3_7/ P5_4 | P3_7/ P5_4 | P3_7/ P5_4 |
| | TAUD0O6 | O | TAUD0 channel output 6 | P5_5 | P5_5 | P5_5 | P5_5 |
| | TAUD0O7 | O | TAUD0 channel output 7 | P3_8/ P5_5 | P3_8/ P5_5 | P3_8/ P5_5 | P3_8/ P5_5 |
| | TAUD0O8 | O | TAUD0 channel output 8 | P5_6 | P5_6 | P5_6 | P5_6 |
| | TAUD0O9 | O | TAUD0 channel output 9 | P3_9/ P5_6 | P3_9/ P5_6 | P3_9/ P5_6 | P3_9/ P5_6 |
| | TAUD0O10 | O | TAUD0 channel output 10 | P0_1 | P0_1 | P0_1/ P5_7 | P0_1/ P5_7 |
| | TAUD0O11 | O | TAUD0 channel output 11 | P3_10 | P3_10 | P3_10/ P5_7 | P3_10/ P5_7 |
| | TAUD0O12 | O | TAUD0 channel output 12 | P5_10 | P5_10 | P5_8/ P5_10 | P5_8/ P5_10 |
| | TAUD0O13 | O | TAUD0 channel output 13 | P3_12 | P3_12 | P3_12/ P5_8 | P3_12/ P5_8 |
| | TAUD0O14 | O | TAUD0 channel output 14 | P5_9 | P5_9 | P5_9 | P5_9 |
| | TAUD0O15 | O | TAUD0 channel output 15 | P3_13/ P5_9 | P3_13/ P5_9 | P3_13/ P5_9 | P3_13/ P5_9 |

Table 2.2 Pin Function (3/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------|-------------------------|------------------------|------------------|-------------------------|-----------------------------------|-----------------------------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| TAUD1 | TAUD1I0 | I | TAUD1 channel input 0 | P3_3/ P2_6 | P3_3/ P2_6 | P3_3/ P2_11/ P2_6 | P3_3/ P2_11/ P2_6 |
| | TAUD1I1 | I | TAUD1 channel input 1 | P2_7 | P2_7 | P2_11/ P2_7 | P2_11/ P2_7 |
| | TAUD1I2 | I | TAUD1 channel input 2 | P3_4/ P2_8 | P3_4/ P2_8 | P3_4/ P2_12/ P2_8 | P3_4/ P2_12/ P2_8 |
| | TAUD1I3 | I | TAUD1 channel input 3 | P2_9 | P2_9 | P2_12/ P2_9 | P2_12/ P2_9 |
| | TAUD1I4 | I | TAUD1 channel input 4 | — | — | P2_13 | P2_13 |
| | TAUD1I5 | I | TAUD1 channel input 5 | — | — | P2_13 | P2_13 |
| | TAUD1I6 | I | TAUD1 channel input 6 | P3_10 | P3_10 | P3_10/ P2_14 | P3_10/ P2_14 |
| | TAUD1I7 | I | TAUD1 channel input 7 | — | — | P2_14 | P2_14 |
| | TAUD1I8 | I | TAUD1 channel input 8 | P3_12 | P3_12 | P3_12/ P2_15 | P3_12/ P2_15 |
| | TAUD1I9 | I | TAUD1 channel input 9 | — | — | P2_15 | P2_15 |
| | TAUD1I10 | I | TAUD1 channel input 10 | P3_13 | P3_13 | P3_13/ P3_0 | P3_13/ P3_0 |
| | TAUD1I11 | I | TAUD1 channel input 11 | — | — | P3_0 | P3_0 |
| | TAUD1I12 | I | TAUD1 channel input 12 | — | — | P3_1 | P3_1 |
| | TAUD1I13 | I | TAUD1 channel input 13 | — | — | P3_1 | P3_1 |
| | TAUD1I14 | I | TAUD1 channel input 14 | — | — | P3_2 | P3_2 |
| | TAUD1I15 | I | TAUD1 channel input 15 | — | — | P3_2 | P3_2 |
| | TAUD1O0 | O | TAUD1 channel output 0 | P3_3/ P2_6 | P3_3/ P2_6 | P3_3/ P2_11/ P2_6/ P4_7 | P3_3/ P2_11/ P2_6/ P4_7 |
| | TAUD1O1 | O | TAUD1 channel output 1 | P3_3/ P2_7 | P3_3/ P2_7 | P3_3/ P2_11/ P2_7/ P4_8 | P3_3/ P2_11/ P2_7/ P4_8 |
| | TAUD1O2 | O | TAUD1 channel output 2 | P3_4/ P2_8 | P3_4/ P2_8 | P3_4/ P2_12/ P2_8/ P4_9 | P3_4/ P2_12/ P2_8/ P4_9 |
| | TAUD1O3 | O | TAUD1 channel output 3 | P3_4/ P2_9 | P3_4/ P2_9 | P3_4/ P2_12/ P2_9/ P4_10 | P3_4/ P2_12/ P2_9/ P4_10 |
| | TAUD1O4 | O | TAUD1 channel output 4 | — | — | P2_13/ P4_11 | P2_13/ P4_11 |
| | TAUD1O5 | O | TAUD1 channel output 5 | — | — | P2_13/ P4_12 | P2_13/ P4_12 |
| | TAUD1O6 | O | TAUD1 channel output 6 | — | — | P2_14/ P4_13 | P2_14/ P4_13 |
| TAUD1O7 | O | TAUD1 channel output 7 | — | — | P2_14/ P4_14 | P2_14/ P4_14 | |
| TAUD1O8 | O | TAUD1 channel output 8 | — | — | P2_15/ P0_3 | P2_15/ P0_3 | |
| TAUD1O9 | O | TAUD1 channel output 9 | — | — | P2_15/ P0_4 | P2_15/ P0_4 | |
| TAUD1O10 | O | TAUD1 channel output 10 | — | — | P3_0/ P0_5/ P5_11 | P3_0/ P0_5/ P5_11 | |
| TAUD1O11 | O | TAUD1 channel output 11 | — | — | P3_0/ P0_6/ P5_12 | P3_0/ P0_6/ P5_12 | |

Table 2.2 Pin Function (4/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------|------------------------|-------------------------|------------------|----------------|--------------------------|--------------------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| TAUD1 | TAUD1O12 | O | TAUD1 channel output 12 | — | — | P3_1/ P0_7/ P5_13 | P3_1/ P0_7/ P5_13 |
| | TAUD1O13 | O | TAUD1 channel output 13 | P5_14 | P5_14 | P3_1/ P0_8/ P5_14 | P3_1/ P0_8/ P5_14 |
| | TAUD1O14 | O | TAUD1 channel output 14 | P0_0 | P0_0 | P3_2/ P0_9/ P0_0 | P3_2/ P0_9/ P0_0 |
| | TAUD1O15 | O | TAUD1 channel output 15 | P0_13/ P0_1 | P0_13/ P0_1 | P3_2/ P0_13/ P0_1 | P3_2/ P0_13/ P0_1 |
| TAUD2 | TAUD2I0 | I | TAUD2 channel input 0 | P0_2/ P4_3 | P0_2/ P4_3 | P0_2/ P4_3 | P0_2/ P4_3 |
| | TAUD2I1 | I | TAUD2 channel input 1 | P4_4 | P4_4 | P0_3/ P4_4 | P0_3/ P4_4 |
| | TAUD2I2 | I | TAUD2 channel input 2 | P4_5 | P4_5 | P0_4/ P4_5 | P0_4/ P4_5 |
| | TAUD2I3 | I | TAUD2 channel input 3 | P4_6 | P4_6 | P0_5/ P4_6 | P0_5/ P4_6 |
| | TAUD2I4 | I | TAUD2 channel input 4 | — | — | P0_6/ P4_7 | P0_6/ P4_7 |
| | TAUD2I5 | I | TAUD2 channel input 5 | — | — | P0_7/ P4_8 | P0_7/ P4_8 |
| | TAUD2I6 | I | TAUD2 channel input 6 | — | — | P0_8/ P4_9 | P0_8/ P4_9 |
| | TAUD2I7 | I | TAUD2 channel input 7 | P0_13 | P0_13 | P0_13/ P0_9/ P4_10 | P0_13/ P0_9/ P4_10 |
| | TAUD2I8 | I | TAUD2 channel input 8 | — | — | P4_11 | P4_11 |
| | TAUD2I9 | I | TAUD2 channel input 9 | — | — | P4_12 | P4_12 |
| | TAUD2I10 | I | TAUD2 channel input 10 | — | — | P2_10/ P4_13 | P2_10/ P4_13 |
| | TAUD2I11 | I | TAUD2 channel input 11 | P2_4/ P2_0 | P2_4/ P2_0 | P2_4/ P2_0/ P4_14 | P2_4/ P2_0/ P4_14 |
| | TAUD2I12 | I | TAUD2 channel input 12 | P2_1/ P1_1 | P2_1/ P1_1 | P2_1/ P1_1 | P2_1/ P1_1 |
| | TAUD2I13 | I | TAUD2 channel input 13 | P2_2/ P1_2 | P2_2/ P1_2 | P2_2/ P1_2 | P2_2/ P1_2 |
| | TAUD2I14 | I | TAUD2 channel input 14 | P2_3/ P1_3 | P2_3/ P1_3 | P2_3/ P1_3 | P2_3/ P1_3 |
| | TAUD2I15 | I | TAUD2 channel input 15 | P1_4 | P1_4 | P1_4 | P1_4 |
| | TAUD2O0 | O | TAUD2 channel output 0 | P0_2/ P4_3 | P0_2/ P4_3 | P0_2/ P4_3 | P0_2/ P4_3 |
| | TAUD2O1 | O | TAUD2 channel output 1 | P4_4 | P4_4 | P0_3/ P4_4 | P0_3/ P4_4 |
| | TAUD2O2 | O | TAUD2 channel output 2 | P4_5 | P4_5 | P0_4/ P4_5 | P0_4/ P4_5 |
| | TAUD2O3 | O | TAUD2 channel output 3 | P4_6 | P4_6 | P0_5/ P4_6 | P0_5/ P4_6 |
| TAUD2O4 | O | TAUD2 channel output 4 | — | — | P0_6/ P4_7 | P0_6/ P4_7 | |
| TAUD2O5 | O | TAUD2 channel output 5 | — | — | P0_7/ P4_8 | P0_7/ P4_8 | |
| TAUD2O6 | O | TAUD2 channel output 6 | — | — | P0_8/ P4_9 | P0_8/ P4_9 | |

Table 2.2 Pin Function (5/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------|----|-------------------------|------------------|----------------|----------------------------------|----------------------------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| TAUD2 | TAUD2O7 | O | TAUD2 channel output 7 | P0_13 | P0_13 | P0_13/ P0_9/ P4_10 | P0_13/ P0_9/ P4_10 |
| | TAUD2O8 | O | TAUD2 channel output 8 | — | — | P4_11 | P4_11 |
| | TAUD2O9 | O | TAUD2 channel output 9 | — | — | P4_12 | P4_12 |
| | TAUD2O10 | O | TAUD2 channel output 10 | — | — | P2_10/ P4_13 | P2_10/ P4_13 |
| | TAUD2O11 | O | TAUD2 channel output 11 | P2_4/ P2_0 | P2_4/ P2_0 | P2_4/ P2_0/ P4_14 | P2_4/ P2_0/ P4_14 |
| | TAUD2O12 | O | TAUD2 channel output 12 | P2_1/ P1_1 | P2_1/ P1_1 | P2_1/ P1_1 | P2_1/ P1_1 |
| | TAUD2O13 | O | TAUD2 channel output 13 | P2_2/ P1_2 | P2_2/ P1_2 | P2_2/ P1_2 | P2_2/ P1_2 |
| | TAUD2O14 | O | TAUD2 channel output 14 | P2_3/ P1_3 | P2_3/ P1_3 | P2_3/ P1_3 | P2_3/ P1_3 |
| | TAUD2O15 | O | TAUD2 channel output 15 | P1_4 | P1_4 | P1_4 | P1_4 |
| TAUJ0 | TAUJ0I0 | I | TAUJ0 channel input 0 | P5_14/ P4_0 | P5_14/ P4_0 | P5_14/ P4_0 | P5_14/ P4_0 |
| | TAUJ0I1 | I | TAUJ0 channel input 1 | P0_0/ P4_1 | P0_0/ P4_1 | P0_0/ P4_1 | P0_0/ P4_1 |
| | TAUJ0I2 | I | TAUJ0 channel input 2 | P0_1/ P4_2 | P0_1/ P4_2 | P0_1/ P4_2 | P0_1/ P4_2 |
| | TAUJ0I3 | I | TAUJ0 channel input 3 | P3_11/ P4_3 | P3_11/ P4_3 | P3_11/ P4_3 | P3_11/ P4_3 |
| | TAUJ0O0 | O | TAUJ0 channel output 0 | P5_14/ P4_0 | P5_14/ P4_0 | P5_14/ P4_0 | P5_14/ P4_0 |
| | TAUJ0O1 | O | TAUJ0 channel output 1 | P0_0/ P4_1 | P0_0/ P4_1 | P0_0/ P4_1 | P0_0/ P4_1 |
| | TAUJ0O2 | O | TAUJ0 channel output 2 | P0_1/ P4_2 | P0_1/ P4_2 | P0_1/ P4_2 | P0_1/ P4_2 |
| | TAUJ0O3 | O | TAUJ0 channel output 3 | P3_11/ P4_3 | P3_11/ P4_3 | P3_11/ P4_3 | P3_11/ P4_3 |
| TAUJ1 | TAUJ1I0 | I | TAUJ1 channel input 0 | P1_1 | P1_1 | P1_1/ P0_5/ P4_11 | P1_1/ P0_5/ P4_11 |
| | TAUJ1I1 | I | TAUJ1 channel input 1 | P1_2 | P1_2 | P1_2/ P0_6/ P4_12 | P1_2/ P0_6/ P4_12 |
| | TAUJ1I2 | I | TAUJ1 channel input 2 | P1_3 | P1_3 | P1_3/ P0_7/ P4_13 | P1_3/ P0_7/ P4_13 |
| | TAUJ1I3 | I | TAUJ1 channel input 3 | P2_4/ P1_4 | P2_4/ P1_4 | P2_4/ P1_4/ P0_8/ P4_14 | P2_4/ P1_4/ P0_8/ P4_14 |
| | TAUJ1O0 | O | TAUJ1 channel output 0 | P1_1 | P1_1 | P1_1/ P0_5/ P4_11 | P1_1/ P0_5/ P4_11 |
| | TAUJ1O1 | O | TAUJ1 channel output 1 | P1_2 | P1_2 | P1_2/ P0_6/ P4_12 | P1_2/ P0_6/ P4_12 |
| | TAUJ1O2 | O | TAUJ1 channel output 2 | P1_3 | P1_3 | P1_3/ P0_7/ P4_13 | P1_3/ P0_7/ P4_13 |
| | TAUJ1O3 | O | TAUJ1 channel output 3 | P2_4/ P1_4 | P2_4/ P1_4 | P2_4/ P1_4/ P0_8/ P4_14 | P2_4/ P1_4/ P0_8/ P4_14 |

Table 2.2 Pin Function (6/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|------------|----------|-------------------------------------|------------------------|----------------|----------------|----------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| TAUJ2 | TAUJ2I0 | I | TAUJ2 channel input 0 | P1_1 | P1_1 | P1_1 | P1_1 |
| | TAUJ2I1 | I | TAUJ2 channel input 1 | P1_2 | P1_2 | P1_2 | P1_2 |
| | TAUJ2I2 | I | TAUJ2 channel input 2 | P1_3 | P1_3 | P1_3 | P1_3 |
| | TAUJ2I3 | I | TAUJ2 channel input 3 | P1_4 | P1_4 | P1_4 | P1_4 |
| | TAUJ2O0 | O | TAUJ2 channel output 0 | P1_1 | P1_1 | P1_1 | P1_1 |
| | TAUJ2O1 | O | TAUJ2 channel output 1 | P1_2 | P1_2 | P1_2 | P1_2 |
| | TAUJ2O2 | O | TAUJ2 channel output 2 | P1_3 | P1_3 | P1_3 | P1_3 |
| | TAUJ2O3 | O | TAUJ2 channel output 3 | P1_4 | P1_4 | P1_4 | P1_4 |
| TSG30 | TSG30O0 | O | TSG30 timer up / down status output | P2_2 | P2_2 | P2_2 | P2_2 |
| | TSG30O1 | O | TSG30 PWM output 1 | P2_3 | P2_3 | P2_3 | P2_3 |
| | TSG30O2 | O | TSG30 PWM output 2 | P2_5 | P2_5 | P2_5 | P2_5 |
| | TSG30O3 | O | TSG30 PWM output 3 | P2_6 | P2_6 | P2_6 | P2_6 |
| | TSG30O4 | O | TSG30 PWM output 4 | P2_7 | P2_7 | P2_7 | P2_7 |
| | TSG30O5 | O | TSG30 PWM output 5 | P2_8 | P2_8 | P2_8 | P2_8 |
| | TSG30O6 | O | TSG30 PWM output 6 | P2_9 | P2_9 | P2_9 | P2_9 |
| | TSG30O7 | O | TSG30 AD trigger diagnostic output | P2_1 | P2_1 | P2_1 | P2_1 |
| | TSG30PTSI0 | I | TSG30 hall sensor input 0 | P4_0 | P4_0 | P4_0 | P4_0 |
| | TSG30PTSI1 | I | TSG30 hall sensor input 1 | P4_1 | P4_1 | P4_1 | P4_1 |
| | TSG30PTSI2 | I | TSG30 hall sensor input 2 | P4_2 | P4_2 | P4_2 | P4_2 |
| | TSG30CLKI | I | TSG30 external clock input enable | P4_3 | P4_3 | P4_3 | P4_3 |
| TSG31 | TSG31O0 | O | TSG31 timer up / down status output | P0_2/ P3_6 | P0_2/ P3_6 | P0_2/ P3_6 | P0_2/ P3_6 |
| | TSG31O1 | O | TSG31 PWM output 1 | P3_7 | P3_7 | P0_3/ P3_7 | P0_3/ P3_7 |
| | TSG31O2 | O | TSG31 PWM output 2 | P3_8 | P3_8 | P0_4/ P3_8 | P0_4/ P3_8 |
| | TSG31O3 | O | TSG31 PWM output 3 | P3_9 | P3_9 | P0_5/ P3_9 | P0_5/ P3_9 |
| | TSG31O4 | O | TSG31 PWM output 4 | P3_10 | P3_10 | P0_6/ P3_10 | P0_6/ P3_10 |
| | TSG31O5 | O | TSG31 PWM output 5 | P3_12 | P3_12 | P0_7/ P3_12 | P0_7/ P3_12 |
| | TSG31O6 | O | TSG31 PWM output 6 | P3_13 | P3_13 | P0_8/ P3_13 | P0_8/ P3_13 |
| | TSG31O7 | O | TSG31 AD trigger diagnostic output | P3_14 | P3_14 | P0_9/ P3_14 | P0_9/ P3_14 |
| | TSG31PTSI0 | I | TSG31 hall sensor input 0 | P4_4 | P4_4 | P0_11/ P4_4 | P0_11/ P4_4 |
| | TSG31PTSI1 | I | TSG31 hall sensor input 1 | P4_5 | P4_5 | P0_12/ P4_5 | P0_12/ P4_5 |
| | TSG31PTSI2 | I | TSG31 hall sensor input 2 | P0_13/ P4_6 | P0_13/ P4_6 | P0_13/ P4_6 | P0_13/ P4_6 |
| | TSG31CLKI | I | TSG31 external clock input enable | P0_0 | P0_0 | P0_0/ P4_7 | P0_0/ P4_7 |
| | ESO | TAPA0ESO | I | Emergency Hi-Z input 0 | P3_14 | P3_14 | P3_14 |
| TAPA1ESO | | I | Emergency Hi-Z input 1 | P0_2/ P4_4 | P0_2/ P4_4 | P0_2/ P4_4 | P0_2/ P4_4 |

Table 2.2 Pin Function (7/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|-----------|----|--|-------------------------|-------------------------|-------------------------|-------------------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| ENCA0 | ENCA0TIN0 | I | ENCA0 capture trigger input 0 | — | — | P4_7 | P4_7 |
| | ENCA0TIN1 | I | ENCA0 capture trigger input 1 | — | — | P4_8 | P4_8 |
| | ENCA0E0 | I | ENCA0 encoder input | P4_0 | P4_0 | P4_0 | P4_0 |
| | ENCA0E1 | I | ENCA0 encoder input | P4_1 | P4_1 | P4_1 | P4_1 |
| | ENCA0EC | I | ENCA0 encoder input | P4_2 | P4_2 | P4_2 | P4_2 |
| ENCA1 | ENCA1TIN0 | I | ENCA1 capture trigger input 0 | — | — | P4_9 | P4_9 |
| | ENCA1TIN1 | I | ENCA1 capture trigger input 1 | — | — | P4_10 | P4_10 |
| | ENCA1E0 | I | ENCA1 encoder input | P4_4 | P4_4 | P0_11/ P4_4 | P0_11/ P4_4 |
| | ENCA1E1 | I | ENCA1 encoder input | P4_5 | P4_5 | P0_12/ P4_5 | P0_12/ P4_5 |
| | ENCA1EC | I | ENCA1 encoder input | P0_13/ P4_6 | P0_13/ P4_6 | P0_13/ P4_6 | P0_13/ P4_6 |
| OSTM0 | OSTM0O | O | OSTM0 timer output | P0_13 | P0_13 | P0_13 | P0_13 |
| OSTM1 | OSTM1O | O | OSTM1 timer output | P2_6/ P4_3 | P2_6/ P4_3 | P2_10/ P2_6/ P4_3 | P2_10/ P2_6/ P4_3 |
| TPBA0 | TPBA0O | O | TPBA0 timer pattern buffer output 0 | P2_2 | P2_2 | P2_2 | P2_2 |
| TPBA1 | TPBA1O | O | TPBA0 timer pattern buffer output 1 | P3_8/ P2_3/ P3_14 | P3_8/ P2_3/ P3_14 | P3_8/ P2_3/ P3_14 | P3_8/ P2_3/ P3_14 |
| CLKDIV0 | EXTCLK0O | O | Clock controller output | P3_14 | P3_14 | P3_14 | P3_14 |
| CLKDIV1 | EXTCLK1O | O | Clock controller output | P4_5 | P4_5 | P4_5 | P4_5 |
| CSIG0 | CSIG0RYI | I | CSIG0 ready (1) / busy (0) input signal | — | — | P3_5 | P3_5 |
| | CSIG0RYO | O | CSIG0 ready (1) / busy (0) output signal | — | — | P3_4 | P3_4 |
| | CSIG0SCI | I | CSIG0 serial clock input signal | P5_4 | P5_4 | P3_2/ P5_4/ P5_2 | P3_2/ P5_4/ P5_2 |
| | CSIG0SCO | O | CSIG0 serial clock output signal | P5_4 | P5_4 | P3_2/ P5_4/ P5_2 | P3_2/ P5_4/ P5_2 |
| CSIG0 | CSIG0SI | I | CSIG0 serial data input | P5_0 | P5_0 | P3_0/ P5_0 | P3_0/ P5_0 |
| | CSIG0SO | O | CSIG0 serial data output | P5_1 | P5_1 | P3_1/ P5_1 | P3_1/ P5_1 |

Table 2.2 Pin Function (8/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|-----------|----|---|------------------|----------------|------------------------------------|------------------------------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| CSIH0 | CSIH0CSS0 | O | CSIH0 serial peripheral chip select signal 0 | P3_12 | P3_12 | P3_12 | P3_12 |
| | CSIH0CSS1 | O | CSIH0 serial peripheral chip select signal 1 | P3_13 | P3_13 | P3_13 | P3_13 |
| | CSIH0CSS2 | O | CSIH0 serial peripheral chip select signal 2 | P2_4 | P2_4 | P2_14/ P2_4 | P2_14/ P2_4 |
| | CSIH0CSS3 | O | CSIH0 serial peripheral chip select signal 3 | P2_5 | P2_5 | P2_13/ P2_5 | P2_13/ P2_5 |
| | CSIH0CSS4 | O | CSIH0 serial peripheral chip select signal 4 | P2_6 | P2_6 | P2_12/ P2_6 | P2_12/ P2_6 |
| | CSIH0CSS5 | O | CSIH0 serial peripheral chip select signal 5 | P2_7/ P2_2 | P2_7/ P2_2 | P2_11/ P2_7/ P2_2 | P2_11/ P2_7/ P2_2 |
| | CSIH0CSS6 | O | CSIH0 serial peripheral chip select signal 6 | P2_8 | P2_8 | P2_10/ P2_8 | P2_10/ P2_8 |
| | CSIH0CSS7 | O | CSIH0 serial peripheral chip select signal 7 | P2_9 | P2_9 | P3_2/ P2_9 | P3_2/ P2_9 |
| | CSIH0SSI | I | CSIH0 serial SS function control input signal | P3_8 | P3_8 | P3_8 | P3_8 |
| | CSIH0RYI | I | CSIH0 ready (1) / busy (0) input signal | P3_9/ P3_3 | P3_9/ P3_3 | P3_9/ P3_3/ P2_15/ P2_10 | P3_9/ P3_3/ P2_15/ P2_10 |
| | CSIH0RYO | O | CSIH0 ready (1) / busy (0) output signal | — | — | P2_11 | P2_11 |
| | CSIH0SCI | I | CSIH0 serial clock input signal | P3_8/ P2_6 | P3_8/ P2_6 | P3_8/ P3_1/ P2_6 | P3_8/ P3_1/ P2_6 |
| | CSIH0SCO | O | CSIH0 serial clock output signal | P3_8/ P2_6 | P3_8/ P2_6 | P3_8/ P3_1/ P2_6 | P3_8/ P3_1/ P2_6 |
| | CSIH0SI | I | CSIH0 serial data input | P3_6/ P2_4 | P3_6/ P2_4 | P3_6/ P3_2/ P2_4 | P3_6/ P3_2/ P2_4 |
| | CSIH0SO | O | CSIH0 serial data output | P3_7/ P2_5 | P3_7/ P2_5 | P3_7/ P3_0/ P2_12/ P2_5 | P3_7/ P3_0/ P2_12/ P2_5 |
| CSIH1 | CSIH1CSS0 | O | CSIH1 serial peripheral chip select signal 0 | P4_3 | P4_3 | P4_3 | P4_3 |
| | CSIH1CSS1 | O | CSIH1 serial peripheral chip select signal 1 | P4_4 | P4_4 | P4_4 | P4_4 |
| | CSIH1CSS2 | O | CSIH1 serial peripheral chip select signal 2 | P4_5 | P4_5 | P4_5 | P4_5 |
| | CSIH1CSS3 | O | CSIH1 serial peripheral chip select signal 3 | P4_6 | P4_6 | P4_6 | P4_6 |
| | CSIH1CSS4 | O | CSIH1 serial peripheral chip select signal 4 | — | — | P4_7 | P4_7 |
| | CSIH1CSS5 | O | CSIH1 serial peripheral chip select signal 5 | — | — | P4_8 | P4_8 |
| | CSIH1CSS6 | O | CSIH1 serial peripheral chip select signal 6 | — | — | P4_9 | P4_9 |
| | CSIH1CSS7 | O | CSIH1 serial peripheral chip select signal 7 | — | — | P4_10 | P4_10 |
| | CSIH1SSI | I | CSIH1 serial SS function control input signal | — | — | P4_8 | P4_8 |
| | CSIH1RYI | I | CSIH1 ready (1) / busy (0) input signal | P3_13 | P3_13 | P3_13/ P2_11/ P4_9 | P3_13/ P2_11/ P4_9 |
| | CSIH1RYO | O | CSIH1 ready (1) / busy (0) output signal | P3_12/ P4_2 | P3_12/ P4_2 | P3_12/ P2_13/ P4_10/ P4_2 | P3_12/ P2_13/ P4_10/ P4_2 |
| | CSIH1SCI | I | CSIH1 serial clock input signal | P2_9/ P4_1 | P2_9/ P4_1 | P2_9/ P4_1 | P2_9/ P4_1 |

Table 2.2 Pin Function (9/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|-----------|-----------|-----------|---|--|----------------------------------|----------------------------------|----------------------------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| CSIH1 | CSIH1SCO | O | CSIH1 serial clock output signal | P2_9/ P4_1 | P2_9/ P4_1 | P2_9/ P4_1 | P2_9/ P4_1 |
| | CSIH1SI | I | CSIH1 serial data input | P2_7/ P3_14 | P2_7/ P3_14 | P2_7/ P3_14 | P2_7/ P3_14 |
| | CSIH1SO | O | CSIH1 serial data output | P2_8/ P4_0 | P2_8/ P4_0 | P2_8/ P4_0 | P2_8/ P4_0 |
| CSIH2 | CSIH2CSS0 | O | CSIH2 serial peripheral chip select signal 0 | P2_3/ P1_1 | P2_3/ P1_1 | P2_3/ P1_1/ P4_7/ P5_2 | P2_3/ P1_1/ P4_7/ P5_2 |
| | CSIH2CSS1 | O | CSIH2 serial peripheral chip select signal 1 | P3_3/ P5_0 | P3_3/ P5_0 | P3_3/ P5_0/ P4_8 | P3_3/ P5_0/ P4_8 |
| | CSIH2CSS2 | O | CSIH2 serial peripheral chip select signal 2 | P3_4/ P5_1 | P3_4/ P5_1 | P3_4/ P5_1/ P4_9 | P3_4/ P5_1/ P4_9 |
| | CSIH2CSS3 | O | CSIH2 serial peripheral chip select signal 3 | P3_5/ P5_4 | P3_5/ P5_4 | P3_5/ P5_4/ P4_10 | P3_5/ P5_4/ P4_10 |
| | CSIH2CSS4 | O | CSIH2 serial peripheral chip select signal 4 | P3_6/ P5_5 | P3_6/ P5_5 | P3_6/ P5_5/ P4_11 | P3_6/ P5_5/ P4_11 |
| | CSIH2CSS5 | O | CSIH2 serial peripheral chip select signal 5 | P3_7/ P5_6 | P3_7/ P5_6 | P3_7/ P5_6/ P4_12 | P3_7/ P5_6/ P4_12 |
| | CSIH2CSS6 | O | CSIH2 serial peripheral chip select signal 6 | P3_8 | P3_8 | P3_8/ P5_7/ P4_14 | P3_8/ P5_7/ P4_14 |
| | CSIH2CSS7 | O | CSIH2 serial peripheral chip select signal 7 | P3_9/ P4_4 | P3_9/ P4_4 | P3_9/ P5_8/ P4_4 | P3_9/ P5_8/ P4_4 |
| | CSIH2SSI | I | CSIH2 serial SS function control input signal | — | — | P4_13/ P5_11 | P4_13/ P5_11 |
| | CSIH2RYI | I | CSIH2 ready (1) / busy (0) input signal | P3_7/ P2_3/ P4_3 | P3_7/ P2_3/ P4_3 | P3_7/ P2_3/ P4_3/ P5_12 | P3_7/ P2_3/ P4_3/ P5_12 |
| | CSIH2RYO | O | CSIH2 ready (1) / busy (0) output signal | P2_4 | P2_4 | P2_4/ P4_7/ P5_13 | P2_4/ P4_7/ P5_13 |
| | CSIH2SCI | I | CSIH2 serial clock input signal | P2_2/ P1_4/ P4_6 | P2_2/ P1_4/ P4_6 | P2_2/ P1_4/ P4_6/ P5_15 | P2_2/ P1_4/ P4_6/ P5_15 |
| | CSIH2SCO | O | CSIH2 serial clock output signal | P2_2/ P1_4/ P4_6 | P2_2/ P1_4/ P4_6 | P2_2/ P1_4/ P4_6/ P5_15 | P2_2/ P1_4/ P4_6/ P5_15 |
| | CSIH2SI | I | CSIH2 serial data input | P2_0/ P1_2/ P4_4/ P5_9 | P2_0/ P1_2/ P4_4/ P5_9 | P2_0/ P1_2/ P4_4/ P5_9 | P2_0/ P1_2/ P4_4/ P5_9 |
| | CSIH2SO | O | CSIH2 serial data output | P2_1/ P1_3/ P4_5/ P5_14 | P2_1/ P1_3/ P4_5/ P5_14 | P2_1/ P1_3/ P4_5/ P5_14 | P2_1/ P1_3/ P4_5/ P5_14 |
| | CSIH3 | CSIH3CSS0 | O | CSIH3 serial peripheral chip select signal 0 | P2_8 | P2_8 | P2_8 |
| CSIH3CSS1 | | O | CSIH3 serial peripheral chip select signal 1 | P2_9 | P2_9 | P2_9 | P2_9 |
| CSIH3CSS2 | | O | CSIH3 serial peripheral chip select signal 2 | P3_5 | P3_5 | P3_5 | P3_5 |
| CSIH3CSS3 | | O | CSIH3 serial peripheral chip select signal 3 | P3_4 | P3_4 | P3_4 | P3_4 |
| CSIH3CSS4 | | O | CSIH3 serial peripheral chip select signal 4 | P3_3 | P3_3 | P3_3 | P3_3 |
| CSIH3CSS5 | | O | CSIH3 serial peripheral chip select signal 5 | P3_9 | P3_9 | P3_9 | P3_9 |

Table 2.2 Pin Function (10/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|-----------|----|---|--|--|--|--|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| CSIH3 | CSIH3CSS6 | O | CSIH3 serial peripheral chip select signal 6 | P2_0 | P2_0 | P2_0 | P2_0 |
| | CSIH3CSS7 | O | CSIH3 serial peripheral chip select signal 7 | P2_1 | P2_1 | P2_1 | P2_1 |
| | CSIH3SSI | I | CSIH3 serial SS function control input signal | P3_3 | P3_3 | P3_3 | P3_3 |
| | CSIH3RYI | I | CSIH3 ready (1) / busy (0) input signal | P3_6/ P2_8/ P1_1 | P3_6/ P2_8/ P1_1 | P3_6/ P2_8/ P1_1 | P3_6/ P2_8/ P1_1 |
| | CSIH3RYO | O | CSIH3 ready (1) / busy (0) output signal | P2_8 | P2_8 | P2_8/ P1_0 | P2_8/ P1_0 |
| | CSIH3SCI | I | CSIH3 serial clock input signal | P2_7/ P1_4 | P2_7/ P1_4 | P2_7/ P1_4 | P2_7/ P1_4 |
| | CSIH3SCO | O | CSIH3 serial clock output signal | P2_7/ P1_4 | P2_7/ P1_4 | P2_7/ P1_4 | P2_7/ P1_4 |
| | CSIH3SI | I | CSIH3 serial data input | P2_6/ P1_2 | P2_6/ P1_2 | P2_6/ P1_2 | P2_6/ P1_2 |
| | CSIH3SO | O | CSIH3 serial data output | P2_5/ P1_3 | P2_5/ P1_3 | P2_5/ P1_3 | P2_5/ P1_3 |
| SCI30 | SCI30RX | I | SCI30 data input | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 | P3_7/ P2_5/ P0_2/ P5_0/ P4_5 |
| | SCI30TX | O | SCI30 data output | P3_12/ P2_6/ P5_1/ P4_6 | P3_12/ P2_6/ P5_1/ P4_6 | P3_12/ P2_6/ P0_3/ P5_1/ P4_6 | P3_12/ P2_6/ P0_3/ P5_1/ P4_6 |
| | SCI30SCI | I | SCI30 serial clock input | P3_13/ P2_7/ P5_4/ P4_2 | P3_13/ P2_7/ P5_4/ P4_2 | P3_13/ P2_7/ P0_4/ P5_4/ P5_2/ P4_2 | P3_13/ P2_7/ P0_4/ P5_4/ P5_2/ P4_2 |
| | SCI30SCO | O | SCI30 serial clock output | P3_13/ P2_7/ P5_4/ P4_2 | P3_13/ P2_7/ P5_4/ P4_2 | P3_13/ P2_7/ P0_4/ P5_4/ P5_2/ P4_2 | P3_13/ P2_7/ P0_4/ P5_4/ P5_2/ P4_2 |
| SCI31 | SCI31RX | I | SCI31 data input | P2_8/ P5_5 | P2_8/ P5_5 | P2_8/ P5_5 | P2_8/ P5_5 |
| | SCI31TX | O | SCI31 data output | P2_9/ P5_6 | P2_9/ P5_6 | P2_9/ P5_6 | P2_9/ P5_6 |
| | SCI31SCI | I | SCI31 serial clock input | P0_13/ P3_5 | P0_13/ P3_5 | P0_13/ P3_5/ P5_7 | P0_13/ P3_5/ P5_7 |
| | SCI31SCO | O | SCI31 serial clock output | P0_13/ P3_5 | P0_13/ P3_5 | P0_13/ P3_5/ P5_7 | P0_13/ P3_5/ P5_7 |
| SCI32 | SCI32RX | I | SCI32 data input | P0_2/ P3_4 | P0_2/ P3_4 | P0_2/ P3_4/ P5_8 | P0_2/ P3_4/ P5_8 |
| | SCI32TX | O | SCI32 data output | P3_3/ P5_9 | P3_3/ P5_9 | P3_3/ P5_9 | P3_3/ P5_9 |
| | SCI32SCI | I | SCI32 serial clock input | P3_9/ P5_10 | P3_9/ P5_10 | P3_9/ P5_10 | P3_9/ P5_10 |
| | SCI32SCO | O | SCI32 serial clock output | P3_9/ P5_10 | P3_9/ P5_10 | P3_9/ P5_10 | P3_9/ P5_10 |
| RLIN30 | RLIN30RX | I | RLIN30 data input | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 | P3_4/ P1_1/ P5_14/ P5_10/ P4_0 |

Table 2.2 Pin Function (11/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|------------|----|--------------------------------------|-------------------------|-------------------------|---|---|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| RLIN30 | RLIN30TX | O | RLIN30 data output | P3_5/ P0_0/ P4_1 | P3_5/ P0_0/ P4_1 | P3_5/ P1_0/ P0_0/ P5_15/ P5_11/ P4_1 | P3_5/ P1_0/ P0_0/ P5_15/ P5_11/ P4_1 |
| RLIN31 | RLIN31RX | I | RLIN31 data input | P2_5/ P0_1 | P2_5/ P0_1 | P2_5/ P0_1/ P5_15/ P5_12 | P2_5/ P0_1/ P5_15/ P5_12 |
| | RLIN31TX | O | RLIN31 data output | P2_4/ P3_11 | P2_4/ P3_11 | P2_4/ P3_11/ P5_13 | P2_4/ P3_11/ P5_13 |
| RSCANFD0 | RSCAN0RX0 | I | RSCAN0 receive data input 0 | P3_7/ P2_0/ P4_5 | P3_7/ P2_0/ P4_5 | P3_7/ P2_0/ P4_5 | P3_7/ P2_0/ P4_5 |
| | RSCAN0TX0 | O | RSCAN0 transmit data output 0 | P3_8/ P2_1/ P4_6 | P3_8/ P2_1/ P4_6 | P3_8/ P2_1/ P4_6 | P3_8/ P2_1/ P4_6 |
| | RSCAN0RX1 | I | RSCAN0 receive data input 1 | P3_12/ P2_2/ P4_2 | P3_12/ P2_2/ P4_2 | P3_12/ P2_2/ P4_7/ P4_2 | P3_12/ P2_2/ P4_7/ P4_2 |
| | RSCAN0TX1 | O | RSCAN0 transmit data output 1 | P3_13/ P2_3/ P4_3 | P3_13/ P2_3/ P4_3 | P3_13/ P2_3/ P4_3 | P3_13/ P2_3/ P4_3 |
| | RSCAN0RX2 | I | RSCAN0 receive data input 2 | P5_6 | P5_6 | P5_6 | P5_6 |
| | RSCAN0TX2 | O | RSCAN0 transmit data output 2 | P5_5 | P5_5 | P5_5/ P5_7 | P5_5/ P5_7 |
| FLXA0 | FLXA0RXDA | I | FLXA0 channel A receive data input | P4_2 | P4_2 | P4_8/ P4_2 | P4_8/ P4_2 |
| | FLXA0RXDB | I | FLXA0 channel B receive data input | P4_3 | P4_3 | P4_11/ P4_3 | P4_11/ P4_3 |
| | FLXA0STPWT | I | FLXA0 stop watch trigger input | P4_4 | P4_4 | P4_14/ P4_4 | P4_14/ P4_4 |
| | FLXA0TXDA | O | FLXA0 channel A transmit data output | P4_0 | P4_0 | P4_9/ P4_0 | P4_9/ P4_0 |
| | FLXA0TXDB | O | FLXA0 channel B transmit data output | P4_5 | P4_5 | P4_12/ P4_5 | P4_12/ P4_5 |
| | FLXA0TXENA | O | FLXA0 channel A transmit enable | P4_1 | P4_1 | P4_10/ P4_1 | P4_10/ P4_1 |
| | FLXA0TXENB | O | FLXA0 channel B transmit enable | P4_6 | P4_6 | P4_13/ P4_6 | P4_13/ P4_6 |
| PSI50 | PSI50DIN | I | PSI50 receive data input | P5_14 | P5_14 | P5_14 | P5_14 |
| | PSI50DOUT | O | PSI50 transmit data output | P0_0 | P0_0 | P0_0 | P0_0 |
| PSI51 | PSI51DIN | I | PSI51 receive data input | P5_9 | P5_9 | P5_9 | P5_9 |
| | PSI51DOUT | O | PSI51 transmit data output | P5_10 | P5_10 | P5_10 | P5_10 |
| SENT0 | SENT0RX | I | SENT ch0 sensor data input | P5_5 | P5_5 | P5_5 | P5_5 |
| | SENT0SPCO | O | SENT ch0 SPC extension output | P5_6/ P5_5 | P5_6/ P5_5 | P0_8/ P5_6/ P5_5 | P0_8/ P5_6/ P5_5 |
| SENT1 | SENT1RX | I | SENT ch1 sensor data input | P5_10 | P5_10 | P5_8/ P5_10 | P5_8/ P5_10 |
| | SENT1SPCO | O | SENT ch1 SPC extension output | P5_9/ P5_10 | P5_9/ P5_10 | P0_7/ P5_9/ P5_8/ P5_10 | P0_7/ P5_9/ P5_8/ P5_10 |
| SENT2 | SENT2RX | I | SENT ch2 sensor data input | — | — | P5_12 | P5_12 |

Table 2.2 Pin Function (12/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|-----------|---------------------|----------------------------------|-------------------------|-------------------------|---|---|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| SENT2 | SENT2SPCO | O | SENT ch2 SPC extension output | — | — | P0_6/ P5_13/ P5_12 | P0_6/ P5_13/ P5_12 |
| SENT3 | SENT3RX | I | SENT ch3 sensor data input | P5_14 | P5_14 | P5_14/ P5_11 | P5_14/ P5_11 |
| | SENT3SPCO | O | SENT ch3 SPC extension output | P0_0/ P5_14 | P0_0/ P5_14 | P0_5/ P0_0/ P5_15/ P5_14/ P5_11 | P0_5/ P0_0/ P5_15/ P5_14/ P5_11 |
| SENT4 | SENT4RX | I | SENT ch4 sensor data input | P0_1 | P0_1 | P0_1 | P0_1 |
| | SENT4SPCO | O | SENT ch4 SPC extension output | P3_11/ P0_1 | P3_11/ P0_1 | P0_4/ P3_11/ P0_1 | P0_4/ P3_11/ P0_1 |
| SENT5 | SENT5RX | I | SENT ch5 sensor data input | P0_2 | P0_2 | P0_2/ P2_10 | P0_2/ P2_10 |
| | SENT5SPCO | O | SENT ch5 SPC extension output | P0_2 | P0_2 | P0_3/ P0_2/ P2_11/ P2_10 | P0_3/ P0_2/ P2_11/ P2_10 |
| ADCG0 | ADCGTRG0 | I | AD trigger input 0 | P5_10 | P5_10 | P0_7/ P5_10 | P0_7/ P5_10 |
| | ADCG0CNV0 | O | ADCG0 AD conversion start signal | P0_2/ P5_10/ P5_1 | P0_2/ P5_10/ P5_1 | P0_2/ P5_10/ P5_1 | P0_2/ P5_10/ P5_1 |
| | ADCG0CNV1 | O | ADCG0 AD conversion start signal | P5_14/ P5_4 | P5_14/ P5_4 | P0_3/ P5_14/ P5_4 | P0_3/ P5_14/ P5_4 |
| | ADCG0CNV2 | O | ADCG0 AD conversion start signal | P0_0/ P5_5 | P0_0/ P5_5 | P0_4/ P0_0/ P5_5 | P0_4/ P0_0/ P5_5 |
| | ADCG0CNV3 | O | ADCG0 AD conversion start signal | P0_1/ P5_6 | P0_1/ P5_6 | P0_5/ P0_1/ P5_6 | P0_5/ P0_1/ P5_6 |
| | ADCG0CNV4 | O | ADCG0 AD conversion start signal | P3_11 | P3_11 | P0_9/ P3_11/ P5_7 | P0_9/ P3_11/ P5_7 |
| | ADCG0I0 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG0I1 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG0I2 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG0I3 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG0I4 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG0I5 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG0I6 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG0I7 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| ADCG0I8 | I | ADCG0 input channel | Not alternative | Not alternative | Not alternative | Not alternative | |
| ADCG0I9 | I | ADCG0 input channel | — | — | Not alternative | Not alternative | |
| ADCG0I10 | I | ADCG0 input channel | — | — | Not alternative | Not alternative | |
| ADCG0I11 | I | ADCG0 input channel | — | — | Not alternative | Not alternative | |

Table 2.2 Pin Function (13/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------------------|----|----------------------------------|------------------|-----------------|-----------------|-----------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| ADCG1 | ADCGTRG1 | I | AD trigger input 1 | P5_14 | P5_14 | P0_8/ P5_14 | P0_8/ P5_14 |
| | ADCG1CNV0 | O | ADCG1 AD conversion start signal | P0_10/ P4_0 | P0_10/ P4_0 | P0_10/ P4_0 | P0_10/ P4_0 |
| | ADCG1CNV1 | O | ADCG1 AD conversion start signal | P4_1 | P4_1 | P0_9/ P4_1 | P0_9/ P4_1 |
| | ADCG1CNV2 | O | ADCG1 AD conversion start signal | P4_2 | P4_2 | P0_11/ P4_2 | P0_11/ P4_2 |
| | ADCG1CNV3 | O | ADCG1 AD conversion start signal | P4_3 | P4_3 | P0_12/ P4_3 | P0_12/ P4_3 |
| | ADCG1CNV4 | O | ADCG1 AD conversion start signal | P4_4 | P4_4 | P0_14/ P4_4 | P0_14/ P4_4 |
| | ADCG1I0 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I1 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I2 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I3 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I4 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I5 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I6 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I7 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I8 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I9 | I | ADCG1 input channel | Not alternative | Not alternative | Not alternative | Not alternative |
| | ADCG1I10 | I | ADCG1 input channel | — | — | Not alternative | Not alternative |
| | ADCG1I11 | I | ADCG1 input channel | — | — | Not alternative | Not alternative |
| Nexus | DCUTCK | I | Debug clock | JP0_2 | JP0_2 | JP0_2 | JP0_2 |
| | DCUTDI | I | Debug data input | JP0_0 | JP0_0 | JP0_0 | JP0_0 |
| | DCUTDO | O | Debug data output | JP0_1 | JP0_1 | JP0_1 | JP0_1 |
| | DCUTMS | I | Debug mode select | JP0_3 | JP0_3 | JP0_3 | JP0_3 |
| | DCUTRDY | O | Debug ready | JP0_5 | JP0_5 | JP0_5 | JP0_5 |
| | DCUTRST ¹ | I | Debug reset | JP0_4 | JP0_4 | JP0_4 | JP0_4 |
| LPD | LPDCLK | I | LPD clock input (4-pin mode) | JP0_2 | JP0_2 | JP0_2 | JP0_2 |
| | LPDCLKOUT | O | LPD clock output (4-pin mode) | JP0_5 | JP0_5 | JP0_5 | JP0_5 |
| | LPDI | I | LPD data input (4-pin mode) | JP0_0 | JP0_0 | JP0_0 | JP0_0 |
| | LPDO | O | LPD data output (4-pin mode) | JP0_1 | JP0_1 | JP0_1 | JP0_1 |
| Nexus | EVTO | O | Debug I/F event output signal | P0_10 | P0_10 | P0_10 | P0_10 |
| | EVTI | I | Debug I/F event input signal | P0_13 | P0_13 | P0_13 | P0_13 |
| FLSCI3 | FLSCI3TXD (FPDT) | O | Flash Writer I/F TxD | JP0_0/ JP0_1 | JP0_0/ JP0_1 | JP0_0/ JP0_1 | JP0_0/ JP0_1 |
| | FLSCI3RXD (FPDR) | I | Flash Writer I/F RxD | JP0_0 | JP0_0 | JP0_0 | JP0_0 |
| | FLSCI3SCKI (FPCK) | I | Flash Writer I/F SCK | JP0_2 | JP0_2 | JP0_2 | JP0_2 |

Table 2.2 Pin Function (14/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|---------------------------------|-----|--|------------------|-----------------|-----------------|-----------------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| MODE | FLMD1 | I | Secondary operating mode select pin | P3_14 | P3_14 | P3_14 | P3_14 |
| System | FLMD0 | I | Primary operating mode select | Not alternative | Not alternative | Not alternative | Not alternative |
| | $\overline{\text{RESET}}$ | I | Reset input | Not alternative | Not alternative | Not alternative | Not alternative |
| | $\overline{\text{RESETOUT}}$ | O | Reset output | P0_10 | P0_10 | P0_10 | P0_10 |
| | X1 | — | Main clock crystal oscillator connections | Not alternative | Not alternative | Not alternative | Not alternative |
| | X2 | — | Main clock crystal oscillator connections | Not alternative | Not alternative | Not alternative | Not alternative |
| CVM | $\overline{\text{CVMOUT}}$ | O | CVM internal voltage error detection output signal | Not alternative | Not alternative | Not alternative | Not alternative |
| Safety | $\overline{\text{ERROROUT}}$ | O | Error output signal | Not alternative | Not alternative | Not alternative | Not alternative |
| | $\overline{\text{ERROROUT_C}}$ | O | Error output signal C | P3_11 | P3_11 | P3_11 | P3_11 |
| JPORT0 | JP0_0 | I/O | JTAG port | JP0_0 | JP0_0 | JP0_0 | JP0_0 |
| | JP0_1 | I/O | JTAG port | JP0_1 | JP0_1 | JP0_1 | JP0_1 |
| | JP0_2 | I | JTAG port | JP0_2 | JP0_2 | JP0_2 | JP0_2 |
| | JP0_3 | I/O | JTAG port | JP0_3 | JP0_3 | JP0_3 | JP0_3 |
| | JP0_4 | I | JTAG port | JP0_4 | JP0_4 | JP0_4 | JP0_4 |
| | JP0_5 | I/O | JTAG port | JP0_5 | JP0_5 | JP0_5 | JP0_5 |
| P0 | P0_0 | I/O | Port | P0_0 | P0_0 | P0_0 | P0_0 |
| | P0_1 | I/O | Port | P0_1 | P0_1 | P0_1 | P0_1 |
| | P0_2 | I/O | Port | P0_2 | P0_2 | P0_2 | P0_2 |
| | P0_3 | I/O | Port | — | — | P0_3 | P0_3 |
| | P0_4 | I/O | Port | — | — | P0_4 | P0_4 |
| | P0_5 | I/O | Port | — | — | P0_5 | P0_5 |
| | P0_6 | I/O | Port | — | — | P0_6 | P0_6 |
| | P0_7 | I/O | Port | — | — | P0_7 | P0_7 |
| | P0_8 | I/O | Port | — | — | P0_8 | P0_8 |
| | P0_9 | I/O | Port | — | — | P0_9 | P0_9 |
| | P0_10 | I/O | Port | P0_10 | P0_10 | P0_10 | P0_10 |
| | P0_11 | I/O | Port | — | — | P0_11 | P0_11 |
| | P0_12 | I/O | Port | — | — | P0_12 | P0_12 |
| | P0_13 | I/O | Port | P0_13 | P0_13 | P0_13 | P0_13 |
| | P0_14 | I/O | Port | — | — | P0_14 | P0_14 |
| P1 | P1_0 | I/O | Port | — | — | P1_0 | P1_0 |
| | P1_1 | I/O | Port | P1_1 | P1_1 | P1_1 | P1_1 |
| | P1_2 | I/O | Port | P1_2 | P1_2 | P1_2 | P1_2 |
| | P1_3 | I/O | Port | P1_3 | P1_3 | P1_3 | P1_3 |
| | P1_4 | I/O | Port | P1_4 | P1_4 | P1_4 | P1_4 |

Table 2.2 Pin Function (15/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------|------|----------|------------------|-------|---------|-------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| P2 | P2_0 | I/O | Port | P2_0 | P2_0 | P2_0 | P2_0 |
| | P2_1 | I/O | Port | P2_1 | P2_1 | P2_1 | P2_1 |
| | P2_2 | I/O | Port | P2_2 | P2_2 | P2_2 | P2_2 |
| | P2_3 | I/O | Port | P2_3 | P2_3 | P2_3 | P2_3 |
| | P2_4 | I/O | Port | P2_4 | P2_4 | P2_4 | P2_4 |
| | P2_5 | I/O | Port | P2_5 | P2_5 | P2_5 | P2_5 |
| | P2_6 | I/O | Port | P2_6 | P2_6 | P2_6 | P2_6 |
| | P2_7 | I/O | Port | P2_7 | P2_7 | P2_7 | P2_7 |
| | P2_8 | I/O | Port | P2_8 | P2_8 | P2_8 | P2_8 |
| | P2_9 | I/O | Port | P2_9 | P2_9 | P2_9 | P2_9 |
| | P2_10 | I/O | Port | — | — | P2_10 | P2_10 |
| | P2_11 | I/O | Port | — | — | P2_11 | P2_11 |
| | P2_12 | I/O | Port | — | — | P2_12 | P2_12 |
| | P2_13 | I/O | Port | — | — | P2_13 | P2_13 |
| | P2_14 | I/O | Port | — | — | P2_14 | P2_14 |
| P2_15 | I/O | Port | — | — | P2_15 | P2_15 | |
| P3 | P3_0 | I/O | Port | — | — | P3_0 | P3_0 |
| | P3_1 | I/O | Port | — | — | P3_1 | P3_1 |
| | P3_2 | I/O | Port | — | — | P3_2 | P3_2 |
| | P3_3 | I/O | Port | P3_3 | P3_3 | P3_3 | P3_3 |
| | P3_4 | I/O | Port | P3_4 | P3_4 | P3_4 | P3_4 |
| | P3_5 | I/O | Port | P3_5 | P3_5 | P3_5 | P3_5 |
| | P3_6 | I/O | Port | P3_6 | P3_6 | P3_6 | P3_6 |
| | P3_7 | I/O | Port | P3_7 | P3_7 | P3_7 | P3_7 |
| | P3_8 | I/O | Port | P3_8 | P3_8 | P3_8 | P3_8 |
| | P3_9 | I/O | Port | P3_9 | P3_9 | P3_9 | P3_9 |
| | P3_10 | I/O | Port | P3_10 | P3_10 | P3_10 | P3_10 |
| | P3_11 | I/O | Port | P3_11 | P3_11 | P3_11 | P3_11 |
| | P3_12 | I/O | Port | P3_12 | P3_12 | P3_12 | P3_12 |
| | P3_13 | I/O | Port | P3_13 | P3_13 | P3_13 | P3_13 |
| | P3_14 | I/O | Port | P3_14 | P3_14 | P3_14 | P3_14 |
| P4 | P4_0 | I/O | Port | P4_0 | P4_0 | P4_0 | P4_0 |
| | P4_1 | I/O | Port | P4_1 | P4_1 | P4_1 | P4_1 |
| | P4_2 | I/O | Port | P4_2 | P4_2 | P4_2 | P4_2 |
| | P4_3 | I/O | Port | P4_3 | P4_3 | P4_3 | P4_3 |
| | P4_4 | I/O | Port | P4_4 | P4_4 | P4_4 | P4_4 |
| | P4_5 | I/O | Port | P4_5 | P4_5 | P4_5 | P4_5 |
| | P4_6 | I/O | Port | P4_6 | P4_6 | P4_6 | P4_6 |
| | P4_7 | I/O | Port | — | — | P4_7 | P4_7 |
| | P4_8 | I/O | Port | — | — | P4_8 | P4_8 |
| | P4_9 | I/O | Port | — | — | P4_9 | P4_9 |
| | P4_10 | I/O | Port | — | — | P4_10 | P4_10 |
| | P4_11 | I/O | Port | — | — | P4_11 | P4_11 |
| P4_12 | I/O | Port | — | — | P4_12 | P4_12 | |

Table 2.2 Pin Function (16/16)

| Category | Pin name | IO | Function | Alternative Port | | | |
|----------|----------|------|----------|------------------|-------|---------|-------|
| | | | | 100-pin | | 144-pin | |
| | | | | eVR | DPS | eVR | DPS |
| P4 | P4_13 | I/O | Port | — | — | P4_13 | P4_13 |
| | P4_14 | I/O | Port | — | — | P4_14 | P4_14 |
| P5 | P5_0 | I/O | Port | P5_0 | P5_0 | P5_0 | P5_0 |
| | P5_1 | I/O | Port | P5_1 | P5_1 | P5_1 | P5_1 |
| | P5_2 | I/O | Port | — | — | P5_2 | P5_2 |
| | P5_3 | I/O | Port | — | — | P5_3 | P5_3 |
| | P5_4 | I/O | Port | P5_4 | P5_4 | P5_4 | P5_4 |
| | P5_5 | I/O | Port | P5_5 | P5_5 | P5_5 | P5_5 |
| | P5_6 | I/O | Port | P5_6 | P5_6 | P5_6 | P5_6 |
| | P5_7 | I/O | Port | — | — | P5_7 | P5_7 |
| | P5_8 | I/O | Port | — | — | P5_8 | P5_8 |
| | P5_9 | I/O | Port | P5_9 | P5_9 | P5_9 | P5_9 |
| | P5_10 | I/O | Port | P5_10 | P5_10 | P5_10 | P5_10 |
| | P5_11 | I/O | Port | — | — | P5_11 | P5_11 |
| | P5_12 | I/O | Port | — | — | P5_12 | P5_12 |
| | P5_13 | I/O | Port | — | — | P5_13 | P5_13 |
| | P5_14 | I/O | Port | P5_14 | P5_14 | P5_14 | P5_14 |
| P5_15 | I/O | Port | — | — | P5_15 | P5_15 | |

Note 1. This is a dedicated pin. Do not use it as a port pin.

2.3 Port Functions

2.3.1 Features

Port Group

This product has the following numbers of port groups:

Table 2.3 RH850/P1M-E Port Group

| Product | Number of Group | Name of Group |
|-------------|-----------------|---------------|
| RH850/P1M-E | 7 | P0 to P5, JP0 |

Port Group Index n

Each port group is identified by its own index “n” (n = 0 to 5) throughout this section; e.g. PMCn for the port mode control register of the Pn pin.

Register Address

All port addresses are given as an offset from the individual base addresses, <JPOR0_base> and <PORT_base>.

Table 2.4 shows the base addresses, <JPOR0_base> and <PORT_base>.

Table 2.4 Port Base Address

| Base address | Address |
|--------------|------------------------|
| <PORT_base> | FFC1 0000 _H |
| <JPOR0_base> | FFC2 0000 _H |

2.3.2 Overview

This product has various pins for input/output (I/O) functions, known as ports. The ports are organized in port groups.

This product also has several control registers to allocate the functions other than general I/O purpose to the corresponding pins.

For definitions of pin, port, and port group, see **Section 2.3.2.1, Terms**.

2.3.2.1 Terms

The terms described in this section are defined as follows.

- **Pin:**
Denotes a physical pin. Every pin is denoted by a unique pin number. Most pins can be used in multiple modes. Thus the pin name indicating its own function is assigned to each pin depending on the selected mode.
- **Port group:**
Denotes a group of ports.
- **Port mode / Port:**
A pin in port mode functions as a general purpose I/O pin. It is then called port.
The corresponding name is Pn_m. For example, P0_7 denotes the port 7 of port group 0 and it is referred to as port P0_7.
- **Alternative mode:**
In alternative mode, a pin can be used for non-general purpose I/O functions as well; e.g. as the I/O pin of on-chip peripherals. Therefore, the corresponding pin name depends on the selected function. For example, the INTPO pin denotes the pin for one of the external interrupt inputs.
Note that two or more different names can refer to the same physical pin, e.g. for P0_2 and INTPO. Those different names indicate the function in which the pin is being operated.
- **Port type:**
The control circuit is selected based on a register setting, and the port type is determined based on the type of the control circuit.

(1) JTAG Ports

The JTAG port groups are used for connecting a debugger for on-chip debugging. These are special port groups provided because the microcontroller cannot be used for the user's application while on-chip debugging is being executed. When a debugger is not connected and the microcontroller is operating normally, these port groups can be used in the same way as the other port groups.

JTAG port group registers and bit names are prefixed by a "J". For example, JP0 denotes JTAG port group 0, and JPM0.JPM0_m denotes the JPM0_m port mode bit of the JPM0 port mode register.

NOTE

In this section, the descriptions about all ports and their registers other than PFCn, PFCAEn, PIPCn, PINVn, and PODCEn apply to the JTAG port unless otherwise specified.

2.3.2.2 Overview of Pin Functions

The pin can operate in the following three different modes:

- Port mode (PMn.PMCn_m = 0)
The pin operates as a general purpose I/O port in port mode.
PMn.PMn_m selects input or output.
- S/W I/O control alternative mode (PMn.PMCn_m = 1, PIPn.PIPCn_m = 0)
The pin is operated by an alternative function in S/W I/O control alternative mode.
The selection between input and output is made by S/W via the PMn.PMn_m control bits.
- Direct I/O control alternative mode (PMn.PMCn_m = 1, PIPn.PIPCn_m = 1)
The pin is operated by an alternative function in direct I/O control alternative mode.
In contrast to S/W I/O control alternative mode, input/output is directly controlled by the alternative function in this mode.

Table 2.5 shows the outline of the register settings.

Table 2.5 Pin Function Configuration (Outline)

| Mode | Bit | | | | I/O |
|-------------------------------------|-------|-------|--------|---------|--|
| | PMn_m | PMn_m | PIPn_m | PIBCn_m | |
| Port mode | 0 | 0 | X | X | O |
| | | 1 | | 1 | I |
| S/W I/O control alternative mode | 1 | 0 | 0 | X | O |
| | | 1 | 0 | 0 | I |
| Direct I/O control alternative mode | | X | 1 | X | Controlled by the alternative function |

- S/W I/O control alternative modes (PIPn.PIPCn_m = 0):
 - Outputs (PMn_m = 0): Alternative output mode 1 to 6
 - Inputs (PMn_m = 1): Alternative input mode 1 to 6
- Direct I/O control alternative modes (PIPn.PIPCn_m = 1):
 - Input/Output of alternative output mode 1 to 6 and alternative input mode 1 to 6 are directly selected by the alternative function.

Table 2.6 Outline of Alternative Mode Selection (PMCn.PMCn_m = 1)

| Function | Register | | | | I/O |
|--------------------------------------|----------|------|-----|------|-----|
| | PFCAE | PFCE | PFC | PM*1 | |
| Alternative output mode 1 (ALT-OUT1) | 0 | 0 | 0 | 0 | O |
| Alternative input mode 1 (ALT-IN1) | 0 | 0 | 0 | 1 | I |
| Alternative output mode 2 (ALT-OUT2) | 0 | 0 | 1 | 0 | O |
| Alternative input mode 2 (ALT-IN2) | 0 | 0 | 1 | 1 | I |
| Alternative output mode 3 (ALT-OUT3) | 0 | 1 | 0 | 0 | O |
| Alternative input mode 3 (ALT-IN3) | 0 | 1 | 0 | 1 | I |
| Alternative output mode 4 (ALT-OUT4) | 0 | 1 | 1 | 0 | O |
| Alternative input mode 4 (ALT-IN4) | 0 | 1 | 1 | 1 | I |
| Alternative output mode 5 (ALT-OUT5) | 1 | 0 | 0 | 0 | O |
| Alternative input mode 5 (ALT-IN5) | 1 | 0 | 0 | 1 | I |
| Alternative output mode 6 (ALT-OUT6) | 1 | 0 | 1 | 0 | O |
| Alternative input mode 6 (ALT-IN6) | 1 | 0 | 1 | 1 | I |

Note 1. When PIPCn.PIPCn_m = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored.

When a pin is operated in alternative mode (PMCn.PMCn_m = 1), one of several alternative functions can be selected by the PFCn, PFCEn, and PFCAEn registers.

2.3.2.3 Pin Data Input/Output

The registers used for data input and output are described below.

The source of the data to be read via the PPRn register depends on pin mode.

Output data

In port mode (PMn.PMCn_m = 0), the value of Pn.Pn_m is output from the Pn_m pin.

Input data

A read operation of the PPRn register returns either the value of the Pn_m pin, the associated bit of the port register Pn.Pn_m, or the data output by an alternative function.

The source of the data read via PPRn depends on pin mode and setting of several control bits.

Table 2.7 summarizes the differences of PPRn read modes.

Table 2.7 PPRn_m Read Values

| PMc n_m | PM n_m | PIBC n_m | PIPC n_m | PODCE n_m | PODC n_m | Mode | PPRn_m read value |
|------------|-----------|-------------|-------------|--------------|-------------|--|--|
| 0 | 1 | 0 | X | X | X | Port input, input buffer disabled | Pn.Pn_m bit |
| | | 1 | | X | X | Port input, input buffer enabled | Pn_m pin |
| | 0 | X | | 0 | 0 | Port push-pull output | Pn.Pn_m bit* ¹ |
| | | | | 0 | 1 | Port N-ch open-drain output | |
| | | | | 1 | 0 | Port push-pull output | |
| | | | | 1 | 1 | Port P-ch open-drain output | |
| 1 | 1 | 0 | 0 | X | X | S/W I/O control alternative input | Pn_m pin |
| | | 1 | | | | Setting prohibited | |
| | 0 | X | 0 | 0 | 0 | S/W I/O control alternative push-pull output | Alternative function internal output signal* ¹ |
| | | | | 0 | 1 | S/W I/O control alternative N-ch open-drain output | |
| | | | | 1 | 0 | S/W I/O control alternative push-pull output | |
| | | | | 1 | 1 | S/W I/O control alternative P-ch open-drain output | |
| | X | 1 | 1 | 0 | 0 | Direct I/O control alternative input or push-pull output | I/O port in alternative function mode: • Input: Pn_m pin • Output: Alternative function internal output signal* ¹ |
| | | | | 0 | 1 | Direct I/O control alternative input or N-ch open-drain output | |
| | | | | 1 | 0 | Direct I/O control alternative input or push-pull output | |
| | | | | 1 | 1 | Direct I/O control alternative input or P-ch open-drain output | |

Note 1. When PBDCn_m = 1, Pn_m pin level is read via PPRn_m.

The control registers in **Table 2.7** have the following effects:

- **PMn.PMCn_m**
This bit selects either port mode ($PMn_m = 0$) or alternative function mode ($PMn_m = 1$).
- **PMn.PMn_m**
This bit selects input ($PMn_m = 1$) or output ($PMn_m = 0$) in port mode ($PMn_m = 0$) and S/W I/O control alternative mode ($PMn_m = 1, PIPCn_m = 0$).
- **PIBCn.PIBCn_m**
This bit disables ($PIBCn_m = 0$) or enables ($PIBCn_m = 1$) the input buffer in input port mode ($PMn_m = 0$ and $PMn_m = 1$). When the input buffer is disabled, $PPRn_m$ reads the $Pn.Pn_m$ bit, otherwise the Pn_m pin level is returned.
- **PIPCn.PIPCn_m**
This bit selects either the S/W or direct I/O control alternative mode.
- **PBDCn.PBDCn_m**
Setting this bit to 1 forces to read the Pn_m pin level via $PPRn_m$. Thus this enables a bidirectional mode, where the level of pin Pn_m can also be read back when the port is operated in an output mode.
- **PODCn.PODCn_m and PODCEn.PODCEn_m bits**
These bits select push-pull output ($PODCn_m$ bit = 0), N-ch open-drain output ($PODCn_m$ bit = 1 and $PODCEn_m$ bit = 0), or P-ch open-drain output ($PODCn_m$ bit = 1 and $PODCEn_m$ bit = 1).

Write to the Pn Register

The data to be output via port Pn_m in port mode ($PMn.PMCn_m = 0$) is held in the port register Pn .

The Pn data can be rewritten in the following two different ways:

- Direct write to the Pn register
New data can be directly written to the Pn register.
- Indirect operation to the Pn bit (set/reset/not)
The indirect Pn operation is possible using the following two registers:
 - Port set/reset register: $PSRn$
When the bit $PSRn.PSRn_{(m+16)} = 1$, the value of bit $PSRn.PSRn_m$ determines the value of $Pn.Pn_m$.
Thus Pn_m can be set/reset without a direct write to Pn .
 - Port NOT register: $PNOTn$
Setting $PNOTn.PNOTn_m = 1$ inverts the bit $Pn.Pn_m$ without a direct write to Pn_m .
Additionally, reading the level of the output pin can read the inverted value of the value that was initially set.

The indirect Pn set/reset/not operation provides access to single bits (not limited to one bit) of the Pn register while leaving all other Pn bits untouched.

2.3.3 Port Type

Figure 2.3 shows the overall configuration of the pins. For the details of port blocks, see Figure 2.4.

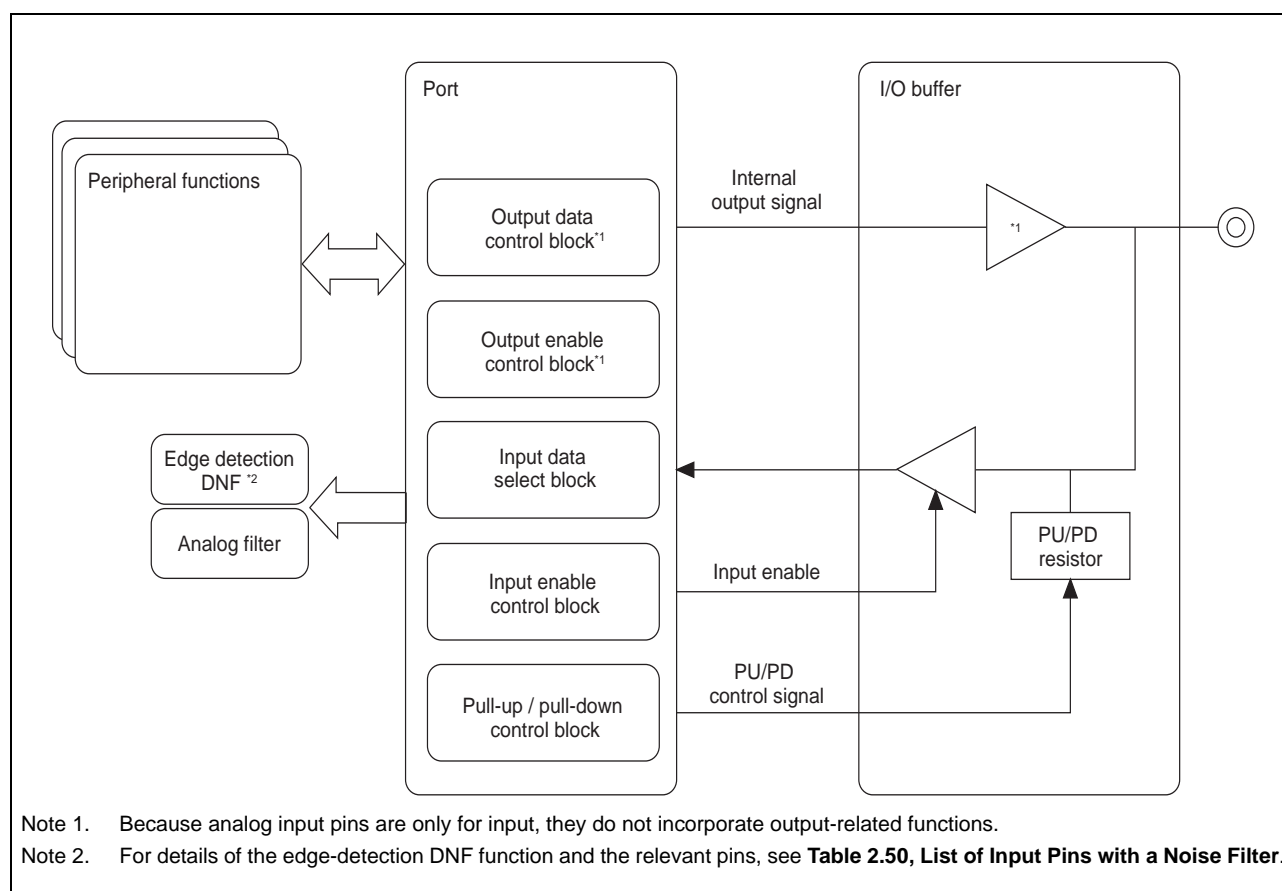


Figure 2.3 Block Diagram of Pin Configuration

Figure 2.4 shows the logical circuitry of the port control functions. The diagram is only a logical reference and does not show the real circuitry.

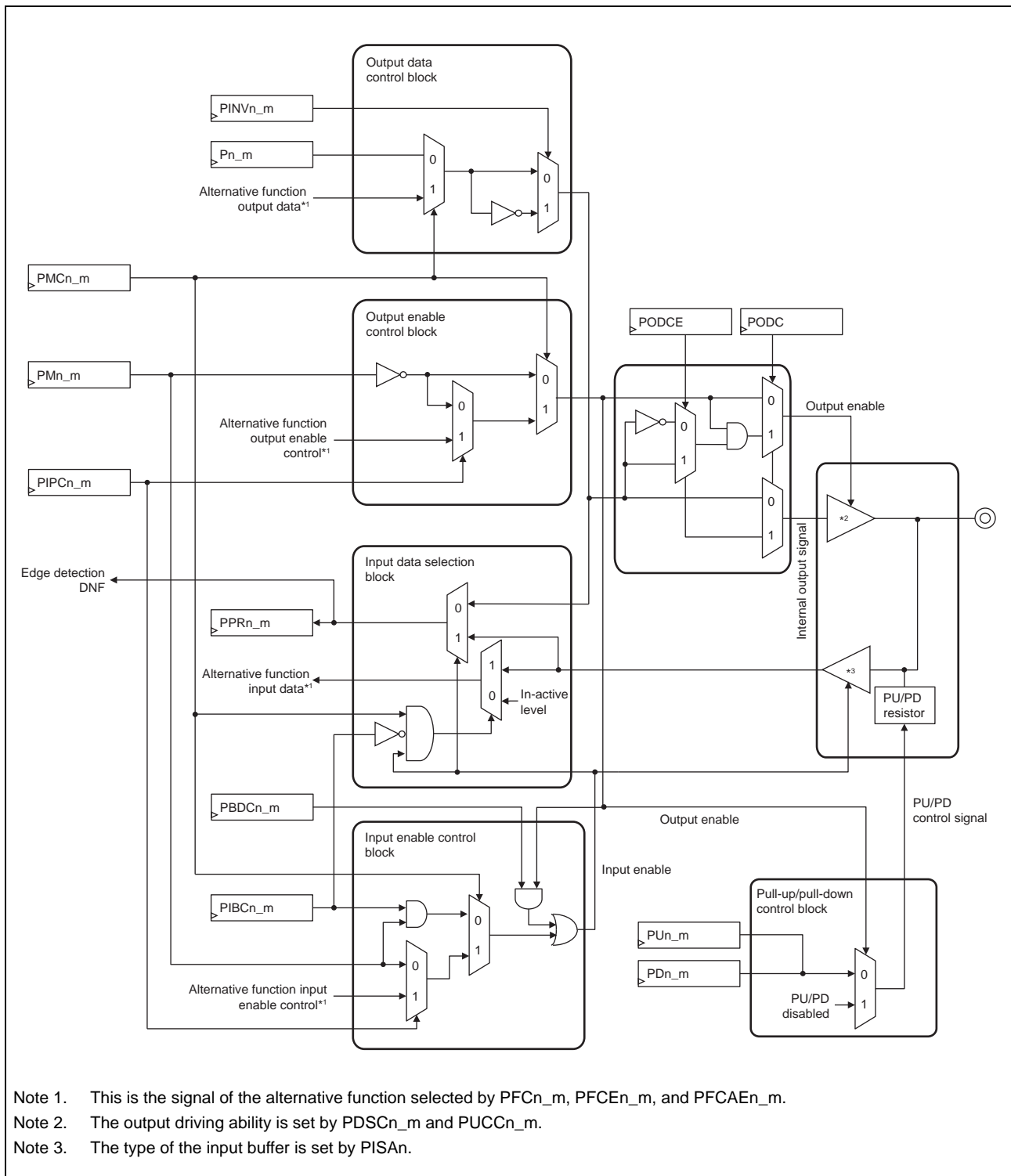


Figure 2.4 Port Control Logic Diagram

2.3.4 Port Group Configuration Register

This section starts with an overview of all configuration registers and then presents all registers in detail. The configuration registers are classified as follows:

- **2.3.4.2 Pin Function Configuration**
- **2.3.4.3 Pin Data Input/Output**
- **2.3.4.4 Configuration of Electrical Characteristics**

2.3.4.1 Outline

The following registers are used for the configuration of the individual pins of the port groups:

Table 2.8 Registers for Port Group Configuration (1/2)

| Register Name | Symbol | Address |
|---|--------------------|--|
| Pin function setting | | |
| Port control register | PCRN _m | <PORT_base> + 2000 _H + n × 40 _H + 4 _H × m |
| | JPCR0 _m | <JPORT0_base> + 2000 _H + 4 _H × m |
| Port mode control register | PMCN | <PORT_base> + 0014 _H + n × 40 _H |
| | JPMC0 | <JPORT0_base> + 0014 _H |
| Port mode control set/reset register | PMCSRn | <PORT_base> + 0024 _H + n × 40 _H |
| | JPMCSR0 | <JPORT0_base> + 0024 _H |
| Port IP control register | PIPCn | <PORT_base> + 4008 _H + n × 40 _H |
| Port mode register | PMn | <PORT_base> + 0010 _H + n × 40 _H |
| | JPM0 | <JPORT0_base> + 0010 _H |
| Port mode set/reset register | PMSRn | <PORT_base> + 0020 _H + n × 40 _H |
| | JPMSR0 | <JPORT0_base> + 0020 _H |
| Port input buffer control register | PIBCn | <PORT_base> + 4000 _H + n × 40 _H |
| | JPIBC0 | <JPORT0_base> + 4000 _H |
| Port function control register | PFCn | <PORT_base> + 0018 _H + n × 40 _H |
| Port function control expansion register | PFCEn | <PORT_base> + 001C _H + n × 40 _H |
| | JPFCE0 | <JPORT0_base> + 001C _H |
| Port function control addition expansion register | PFCAEn | <PORT_base> + 0028 _H + n × 40 _H |
| Pin data input/output | | |
| Port bi-direction control register | PBDCn | <PORT_base> + 4004 _H + n × 40 _H |
| | JPBDC0 | <JPORT0_base> + 4004 _H |
| Port pin read register | PPRn | <PORT_base> + 000C _H + n × 40 _H |
| | JPPR0 | <JPORT0_base> + 000C _H |
| Port register | Pn | <PORT_base> + 0000 _H + n × 40 _H |
| | JP0 | <JPORT0_base> + 0000 _H |
| Port NOT register | PNOTn | <PORT_base> + 0008 _H + n × 40 _H |
| | JPNOT0 | <JPORT0_base> + 0008 _H |
| Port set/reset register | PSRn | <PORT_base> + 0004 _H + n × 40 _H |
| | JPSR0 | <JPORT0_base> + 0004 _H |
| Port output level inversion register | PINVn | <PORT_base> + 0030 _H + n × 40 _H |

Table 2.8 Registers for Port Group Configuration (2/2)

| Register Name | Symbol | Address |
|--|--------|---|
| Specifying electrical characteristics | | |
| Pull-up option register | PUn | <PORT_base> + 400C _H + n × 40 _H |
| | JPU0 | <JPORT0_base> + 400C _H |
| Pull-down option register | PDn | <PORT_base> + 4010 _H + n × 40 _H |
| | JPD0 | <JPORT0_base> + 4010 _H |
| Port drive strength control register | PDSCn | <PORT_base> + 4018 _H + n × 40 _H |
| | JPDSC0 | <JPORT0_base> + 4018 _H |
| Port open-drain control register | PODCn | <PORT_base> + 4014 _H + n × 40 _H |
| | JPODC0 | <JPORT0_base> + 4014 _H |
| Port open-drain control expansion register | PODCEn | <PORT_base> + 403C _H + n × 40 _H |
| Port universal control register | PUCCn | <PORT_base> + 4028 _H + n × 40 _H |
| | JPUCC0 | <JPORT0_base> + 4028 _H |
| Port input buffer selection register | PISAn | <PORT_base> + 402C _H + n × 40 _H |
| | JPISA0 | <JPORT0_base> + 402C _H |

Note: n: Port group number
m: Bit number in a port group

Base address

The PORTn base addresses, <PORT_Base> and <JPORT0_Base> are defined in **Register Address, Section 2.3.1, Features**.

Value after a reset

The value after a reset depends on the port, and are not described in the following register descriptions, but are given in **Section 2.4.1.1, List of Port Registers**.

2.3.4.2 Pin Function Configuration

(1) PMCn/JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

Access: PMCn: This register can be read/written in 16-bit units.
JPMC0: This register can be read/written in 8-bit units.

Address: PMCn: <PORT_base> + 0014_H + n × 40_H
JPMC0: <JPORT0_base> + 0014_H*1

Value after reset: PMCn: 0000_H
JPMC0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | PMCn_15 | PMCn_14 | PMCn_13 | PMCn_12 | PMCn_11 | PMCn_10 | PMCn_9 | PMCn_8 | PMCn_7 | PMCn_6 | PMCn_5 | PMCn_4 | PMCn_3 | PMCn_2 | PMCn_1 | PMCn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.9 PMCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 15 to 0 | PMCn_[15:0] | Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode |

CAUTIONS

- The I/O control is not performed by just setting alternative mode (PMCn.PMCn_m). When the alternative function performs direct I/O control, set also the PIPcn.PIPcn_m bit to 1.
- When a port is used as an input pin in alternative mode, there is a pin that passes a noise filter. Such a pin might need FCLA0CTLm.DFNAnCTL and the DFNAnEN register to be set. For details, see **Section 2.6, Noise Filter and Edge Level Detection Circuit**.

NOTES

- The control bits of the JTAG port mode control register (JPMC0) are JPMC0_[7:0].
- For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers**.

(2) PMCSRn/JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method to write data to a bit in the PMCN register.

The upper 16 bits of PMCSRn specify whether the data of the corresponding lower 16 bits of PMCSRn is written to the PMCN.PMCn_m bit.

Access: This register can be read/written in 32-bit units.

PMCSRn: Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of the PMCN register.

JPMCSR: Bits 31 to 8 are always read as 0000 00_H. Reading bits 7 to 0 returns the value of the JPMCN register.

Address: PMCSRn: <PORT_base> + 0024_H + n × 40_H
JPMCSR0: <JPORT0_base> + 0024_H*1

Value after reset: 0000 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | PMCSR n_31 | PMCSR n_30 | PMCSR n_29 | PMCSR n_28 | PMCSR n_27 | PMCSR n_26 | PMCSR n_25 | PMCSR n_24 | PMCSR n_23 | PMCSR n_22 | PMCSR n_21 | PMCSR n_20 | PMCSR n_19 | PMCSR n_18 | PMCSR n_17 | PMCSR n_16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PMCSR n_15 | PMCSR n_14 | PMCSR n_13 | PMCSR n_12 | PMCSR n_11 | PMCSR n_10 | PMCSR n_9 | PMCSR n_8 | PMCSR n_7 | PMCSR n_6 | PMCSR n_5 | PMCSR n_4 | PMCSR n_3 | PMCSR n_2 | PMCSR n_1 | PMCSR n_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.10 PMCSRn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|--|
| 31 to 16 | PMCSRn_ [31:16] | Specifies whether the value of the corresponding lower bit of PMCSRn_[15:0] is written to PMCN_m: 0: PMCN_m does not depend on PMCSRn_m. 1: The value of PMCN_m is the same as that of PMCSRn_m. Example: When PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCN.PMCn_15. |
| 15 to 0 | PMCSRn_ [15:0] | Specifies the PMCN_m value when the corresponding upper bit PMCSRn_(m+16) is 1: 0: PMCN_m = 0 1: PMCN_m = 1 |

NOTES

- The control bits of the JTAG port mode control set/reset register (JPMCSR0) are JPMCSR0_[31:0].
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(3) PIPCN — Port IP Control Register

This register specifies whether the I/O direction of pin Pn_m is controlled by the port mode register PMn.PMn_m or by an alternative function.

When the Pn_m pin is operated in alternative mode (PMcn.PMCn_m = 1) and the alternative function directly controls the I/O direction of Pn_m, PIPCN.PIPCn_m must be set to 1 as well.

This hands over I/O control to the alternative function and overrules the PMn.PMn_m setting.

Access: This register can be read/written in 16-bit units.

Address: PIPCN: <PORT_base> + 4008_H + n × 40_H^{*1}

Value after reset: 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| | PIPCn_15 | PIPCn_14 | PIPCn_13 | PIPCn_12 | PIPCn_11 | PIPCn_10 | PIPCn_9 | PIPCn_8 | PIPCn_7 | PIPCn_6 | PIPCn_5 | PIPCn_4 | PIPCn_3 | PIPCn_2 | PIPCn_1 | PIPCn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.11 PIPCN Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 15 to 0 | PIPCn_[15:0] | Specifies the I/O control mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control). |

NOTE

For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(4) PMn/JPM0 — Port Mode Register

The PMn register specifies whether the individual pins of port group n are in input mode or in output mode.

Access: PMn: This register can be read/written in 16-bit units.
JPM0: This register can be read/written in 8-bit units.

Address: PMn: <PORT_base> + 0010_H + n × 40_H
JPM0: <JPORT0_base> + 0010_H*1

Value after reset: PM0: FBFF_H, PM1 to 5: FFFF_H
JPM0: FF_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|--------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | PMn_1 5 | PMn_1 4 | PMn_1 3 | PMn_1 2 | PMn_11 | PMn_1 0 | PMn_9 | PMn_8 | PMn_7 | PMn_6 | PMn_5 | PMn_4 | PMn_3 | PMn_2 | PMn_1 | PMn_0 |
| Value after reset | 1 | 1 | 1 | 1 | 1 | *1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. PM0: 0
PM1 to PM5: 1

Table 2.12 PMn Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 15 to 0 | PMn_[15:0] | Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled) |

NOTES

- When a port is used in input port mode (PMnCn.PMCn_m = 0 and PMn.PMn_m = 1), the input buffer needs to be enabled (PIBCn.PIBCn_m = 1).
- After reset, PIPCN.PIPCN_m = 0 is set (I/O mode is controlled with PMn.PMn_m). Thus, PMn_m specifies the I/O direction of port mode (PMnCn.PMCn_m = 0) and alternative mode (PMnCn.PMCn_m = 1).
- The control bits of the JTAG port mode register (JPM0) are JPM0_[7:0].
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(5) PMSRn/JPMSR0 — Port Mode Set/Reset Register n

This register provides an alternative method to write data to a bit in the PMn register.

The upper 16 bits of PMSRn specify whether the data of the corresponding lower 16 bits of PMSRn is written to the PMn.PMn_m bit.

Access: This register can be read/written in 32-bit units.
PMSRn: Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of the PMn register.
JPMSRn: Bits 31 to 8 are always read as 0000 00_H. Reading bits 7 to 0 returns the value of the JPMCn register.

Address: PMSRn: <PORT_base> + 0020_H + n × 40_H
JPMSR0: <JPORT0_base> + 0020_H^{*1}

Value after reset: PMSR0: 0000 FBFF_H, PMSR1 to 5: 0000 FFFF_H
JPMSR0: 0000 00FF_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | PMSRn _31 | PMSRn _30 | PMSRn _29 | PMSRn _28 | PMSRn _27 | PMSRn _26 | PMSRn _25 | PMSRn _24 | PMSRn _23 | PMSRn _22 | PMSRn _21 | PMSRn _20 | PMSRn _19 | PMSRn _18 | PMSRn _17 | PMSRn _16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PMSRn _15 | PMSRn _14 | PMSRn _13 | PMSRn _12 | PMSRn _11 | PMSRn _10 | PMSRn _9 | PMSRn _8 | PMSRn _7 | PMSRn _6 | PMSRn _5 | PMSRn _4 | PMSRn _3 | PMSRn _2 | PMSRn _1 | PMSRn _0 |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *2 | *1 | *1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. PMSR0 to PMSR5: 1
JPMSR0: 0

Note 2. PMSR1 to PMSR5: 1
PMSR0, JPMSR0: 0

Table 2.13 PMSRn Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31 to 16 | PMSRn_[31:16] | Specifies whether the value of the corresponding lower bit of PMSRn_[15:0] is written to PMn_m: 0: PMn_m does not depend on PMSRn_m. 1: The value of PMn_m is the same as that of PMSRn_m. Example: When PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15 and output. |
| 15 to 0 | PMSRn_[15:0] | Specifies the PMn_m value when the corresponding upper bit PMSRn_(m+16) is 1: 0: PMn_m = 0 1: PMn_m = 1 |

NOTES

- The control bits of the JTAG port mode set/reset register (JPMSR0) are JPMSR0_[31:0]
- For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers.**

(6) PIBCn/JPIBC0 — Port Input Buffer Control Register

This register enables and disables the input buffer in input port mode (PMnCn.PMCn_m = 0 and PMn.PMn_m = 1).

Additionally, when pins are in bidirectional mode (PBDCn.PBDCn_m = 1), it is capable of selecting shared output level loop-back function and pin output level-read function with the setting of PIBCn.PIBCn_m. For details, see **(1) PBDCn/JPBDC0 — Port Bi-Direction Control Register** in **Section 2.3.4.3, Pin Data Input/Output**.

Access: PIBCn: This register can be read/written in 16-bit units.
JPIBC0: This register can be read/written in 8-bit units.

Address: PIBCn: <PORT_base> + 4000_H + n × 40_H
JPIBC0: <JPORT0_base> + 4000_H*1

Value after reset: PIBCn: 0000_H
JPIBC0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| | PIBCn_15 | PIBCn_14 | PIBCn_13 | PIBCn_12 | PIBCn_11 | PIBCn_10 | PIBCn_9 | PIBCn_8 | PIBCn_7 | PIBCn_6 | PIBCn_5 | PIBCn_4 | PIBCn_3 | PIBCn_2 | PIBCn_1 | PIBCn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.14 PIBCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 15 to 0 | PIBCn_[15:0] | Enables/disables the input buffer: 0: Input buffer is disabled. 1: Input buffer is enabled. |

NOTES

- When the input buffer is disabled, it does not consume flow-through current even if the pin level is in Hi-Z state. Thus, the pin does not need to be fixed to a high or low level externally.
- The control bits of the JTAG port input buffer control register (JPIBC0) are JPIBC0_[7:0].
- For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers**.

(7) PFCn — Port Function Control Register

This register, together with the PFCEn and PFCAEn registers, specifies an alternative function of the pins.

Some alternative functions require direct I/O control of pin Pn_m. For such alternative functions PIPCN.PIPCN_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: This register can be read/written in 16-bit units.

Address: <PORT_base> + 0018_H + n × 40_H^{*1}

Value after reset: 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | PFCn_15 | PFCn_14 | PFCn_13 | PFCn_12 | PFCn_11 | PFCn_10 | PFCn_9 | PFCn_8 | PFCn_7 | PFCn_6 | PFCn_5 | PFCn_4 | PFCn_3 | PFCn_2 | PFCn_1 | PFCn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.15 PFCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 15 to 0 | PFCn_[15:0] | Specifies an alternative function of a pin. See Table 2.6, Outline of Alternative Mode Selection (PMcn.PMCn_m = 1). |

NOTE

For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(8) PFCEn / JPFCE0 — Port Function Control Expansion Register

This register, together with the PFCn and PFCAEn registers, specifies an alternative function of the pins.

Some alternative functions directly controls input/output of pin Pn_m. For such alternative functions PIPCn.PIPCn_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: PFCEn: This register can be read/written in 16-bit units.
JPFCE0: This register can be read/written in 8-bit units.

Address: PFCEn: <PORT_base> + 001C_H + n × 40_H
JPFCE0: <JPORT0_base> + 001C_H¹

Value after reset: PFCEn: 0000_H
JPFCE0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | PFCEn _15 | PFCEn _14 | PFCEn _13 | PFCEn _12 | PFCEn _11 | PFCEn _10 | PFCEn _9 | PFCEn _8 | PFCEn _7 | PFCEn _6 | PFCEn _5 | PFCEn _4 | PFCEn _3 | PFCEn _2 | PFCEn _1 | PFCEn _0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.16 PFCEn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 15 to 0 | PFCEn_[15:0] | Specifies an alternative function of a pin. See Table 2.6, Outline of Alternative Mode Selection (PMcN.PMcN_m = 1) for details. |

NOTES

1. The control bits of the JTAG port function control expansion register (JPFCE0) are JPFCE0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(9) PFCAEn — Port Function Control Additional Expansion Register

This register, together with the PFCn and PFCEn registers, specifies an alternative function of the pins. Some alternative functions directly controls input/output of pin Pn_m. For such an alternative functions PIPCn.PIPCn_m must be set to 1 as well. For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: This register can be read/written in 16-bit units.

Address: <PORT_base> + 0028_H + n × 40_H^{*1}

Value after reset: 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | PFCAE n_15 | PFCAE n_14 | PFCAE n_13 | PFCAE n_12 | PFCAE n_11 | PFCAE n_10 | PFCAE n_9 | PFCAE n_8 | PFCAE n_7 | PFCAE n_6 | PFCAE n_5 | PFCAE n_4 | PFCAE n_3 | PFCAE n_2 | PFCAE n_1 | PFCAE n_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.17 PFCAEn Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 15 to 0 | PFCAEn_[15:0] | Specifies an alternative function of a pin. See Table 2.6, Outline of Alternative Mode Selection (PMcN.PMcN_m = 1) for details. |

Table 2.18 Setting Alternative Functions

| PFCAEn_m | PFCEn_m | PFCn_m | PMn_m | Function | | |
|----------|---------|--------|-------|-----------------------------------|-----------------------------------|-----------------------------------|
| 0 | 0 | 0 | 1 | 1st alternative function / Input | | |
| | | | 0 | 1st alternative function / Output | | |
| | | | 1 | 1 | 2nd alternative function / Input | |
| | | | | 0 | 2nd alternative function / Output | |
| | | | 1 | 0 | 1 | 3rd alternative function / Input |
| | | | | | 0 | 3rd alternative function / Output |
| 1 | 0 | 0 | 1 | 4th alternative function / Input | | |
| | | | 0 | 4th alternative function / Output | | |
| | | | 1 | 1 | 5th alternative function / Input | |
| | | | | 0 | 5th alternative function / Output | |
| | | | 1 | 1 | 1 | 6th alternative function / Input |
| | | | | | 0 | 6th alternative function / Output |
| 1 | 1 | X | X | Setting prohibited | | |

CAUTION

- After selecting the alternative function by the PFCn_m, PFCEn_m or PFCAEn_m register, set the PMcN_m register to 1.
- With this product, the I/O of some functions is multiplexed in two pins, but only either one can be used as a pin function. Setting the same pin function in two pins is prohibited. For example, if the a/b/c pin is used as b, the b/d/e pin cannot be used as b. In this case, the b/d/e pin must be configured as the pin function other than b.

NOTE

For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers**.

2.3.4.3 Pin Data Input/Output

(1) PBDCn/JPBDC0 — Port Bi-Direction Control Register

This register enables the input buffer when a port is used in output mode and sets in bi-direction mode. In bi-direction mode, PPRn.PPRn_m can read the level of the Pn_m pin.

- **Alternative output level loopback function**
When the Pn_m pin is used as the alternative output function, the actual pin output level based on the alternative output function can be looped back to the alternative input side by setting PBDCn.PBDCn_m = 1 and PIBCn.PIBCn_m = 0. For example, the pin output level based on the first alternative function can be looped back to the same alternative input side. Also the pin output level can be read via PPRn.PPRn_m.
- **Pin output level read function**
When the Pn_m pin is used as the general output port function or the alternative output function, the actual pin output level can be read via PPRn.PPRn_m by setting PBDCn.PBDCn_m = 1 and PIBCn.PIBCn_m = 1. Under this setting, the pin output level will never be looped back to the alternative input side even in alternative output mode.

Access: PBDCn: This register can be read/written in 16-bit units.
JPBDC0: This register can be read/written in 8-bit units.

Address: PBDCn: <PORT_base> + 4004_H + n × 40_H
JPBDC0: <JPOR0_base> + 4004_H^{*1}

Value after reset: PBDCn: 0000_H
JPBDC0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | PBDCn _15 | PBDCn _14 | PBDCn _13 | PBDCn _12 | PBDCn _11 | PBDCn _10 | PBDCn _9 | PBDCn _8 | PBDCn _7 | PBDCn _6 | PBDCn _5 | PBDCn _4 | PBDCn _3 | PBDCn _2 | PBDCn _1 | PBDCn _0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.19 PBDCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 15 to 0 | PBDCn_[15:0] | Enables/disables bi-direction mode of the corresponding pin. 0: Bi-direction mode disabled 1: Bi-direction mode enabled |

CAUTION

When the Pn_m port is used as an alternative output function (PMc.PMc_m = 1, PMn.PMn_m = 0), if bidirectional mode is enabled (PBDCn.PBDCn_m = 1), the level of the Pn_m pin can be read with PPRn.PPRn_m.

However, the output of the alternative output function is input to the alternative input function of the same pin (the alternative input function that is set with PFCn.PFCn_m, PFCEn.PFCEn_m, and PFCAEn.PFCAEn_m). If this alternative input function is used in another pin, note that the output of the alternative output function and alternative input in other pins are input internally (OR input).

NOTES

1. The control bits of the JTAG port bi-direction control register (JPBDC0) are JPBDC0_[7:0].
2. For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers**.

(2) PPRn/JPPR0 — Port Pin Read Register

This register reflects an actual Pn_m pin level, a Pn.Pn_m bit value, or an output level of the alternative function. The value to be read depends on various control settings as described in **Table 2.7, PPRn_m Read Values**.

Access: PPRn: This register can be read only in 16-bit units.
JPPR0: This register can only be read in 8-bit units.

Address: PPRn: <PORT_base> + 000C_H + n × 40_H
JPPR0: <JPORT0_base> + 000C_H*1

Value after reset: PPRn: 0000_H
JPPR0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (**Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0**) in **Section 2.4.1.1, List of Port Registers**.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | PPRn_15 | PPRn_14 | PPRn_13 | PPRn_12 | PPRn_11 | PPRn_10 | PPRn_9 | PPRn_8 | PPRn_7 | PPRn_6 | PPRn_5 | PPRn_4 | PPRn_3 | PPRn_2 | PPRn_1 | PPRn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 2.20 PPRn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 15 to 0 | PPRn_[15:0] | Pin Pn_m, Pn.Pn_m value or alternative function output. |

NOTES

1. For details about the read value of the PPRn register, see **Section 2.3.2.3, Pin Data Input/Output**.
2. The control bits of the JTAG port pin read register (JPPR0) are JPPR0_[7:0].
3. For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers**.

(3) Pn/JP0 — Port Register

This register sets and holds the Pn.Pn_m data to be output via the related Pn_m port in output port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 0).

Access: Pn: This register can be read/written in 16-bit units.
JP0: This register can be read/written in 8-bit units.

Address: Pn: <PORT_base> + 0000_H + n × 40_H
JP0: <JP0_base> + 0000_H*¹

Value after reset: Pn: 0000_H
JP0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| | Pn_15 | Pn_14 | Pn_13 | Pn_12 | Pn_11 | Pn_10 | Pn_9 | Pn_8 | Pn_7 | Pn_6 | Pn_5 | Pn_4 | Pn_3 | Pn_2 | Pn_1 | Pn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.21 Pn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 15 to 0 | Pn_[15:0] | Sets the output level of pin m (m = 0 to 15): 0: Low level output 1: High level output |

NOTES

- The bits of this register can be manipulated by various means; refer to the subsection, **Write to the Pn Register** in Section 2.3.2.3, Pin Data Input/Output.
- The control bits of the JTAG port register (JP0) are JP0_[7:0].
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(4) PNOTn/JPNOT0 — Port NOT Register

This register enables inverting the Pn_m bit of the port register without directly writing to Pn.

Access: PNOTn: This register can be written in 16-bit units. The read value is always 0000_H.
JPNOT0: This register can only be written in 8-bit units. The read value is always 00_H.

Address: PNOTn: <PORT_base> + 0008_H + n × 40_H
JPNOT0: <JPORT0_base> + 0008_H^{*1}

Value after reset: PNOTn: 0000_H
JPNOT0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | PNOTn _15 | PNOTn _14 | PNOTn _13 | PNOTn _12 | PNOTn _11 | PNOTn _10 | PNOTn _9 | PNOTn _8 | PNOTn _7 | PNOTn _6 | PNOTn _5 | PNOTn _4 | PNOTn _3 | PNOTn _2 | PNOTn _1 | PNOTn _0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 2.22 PNOTn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 15 to 0 | PNOTn_[15:0] | Specifies if Pn.Pn_m is inverted or not: 0: Pn.Pn_m is not inverted (Pn_m → Pn_m) 1: Pn.Pn_m is inverted ($\overline{Pn_m}$ → Pn_m) |

NOTES

1. The control bits of the JTAG port not register (JPNOT0) are JPNOT0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(5) PSRn/JPSR0 — Port Set/Reset Register

This register provides an alternative method to write data to a bit in the Pn register.

The upper 16 bits of PSRn specify whether the data of the corresponding lower 16 bits of PSRn is written to the Pn.Pn_m bit.

Access: This register can be read/written in 32-bit units.
 PSRn: Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of the PMCN register.
 JPSRn: Bits 31 to 8 are always read as 0000 00_H. Reading bits 15 to 0 returns the value of the JPn register.

Address: PSRn: <PORT_base> + 0004_H + n × 40_H
 JPSR0: <JPORT0_base> + 0004_H^{*1}

Value after reset: PSRn: 0000 0000_H
 JPSR0: 0000 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| | PSRn_31 | PSRn_30 | PSRn_29 | PSRn_28 | PSRn_27 | PSRn_26 | PSRn_25 | PSRn_24 | PSRn_23 | PSRn_22 | PSRn_21 | PSRn_20 | PSRn_19 | PSRn_18 | PSRn_17 | PSRn_16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PSRn_15 | PSRn_14 | PSRn_13 | PSRn_12 | PSRn_11 | PSRn_10 | PSRn_9 | PSRn_8 | PSRn_7 | PSRn_6 | PSRn_5 | PSRn_4 | PSRn_3 | PSRn_2 | PSRn_1 | PSRn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.23 PSRn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 16 | PSRn_[31:16] | Specifies whether the value of the corresponding lower bit of PSRn_[15:0] is written to Pn_m: 0: Pn_m does not depend on PSRn_m. 1: The value of Pn_m is the same as that of PSRn_m. Example: When PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15. |
| 15 to 0 | PSRn_[15:0] | Specifies the Pn_m value when the corresponding upper bit PSRn_(m+16) is 1: 0: Pn_m = 0 1: Pn_m = 1 |

NOTES

- The control bits of the JTAG port set/reset register (JPSR0) are JPSR0_[31:0].
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(6) PINVn — Port Output Level Inversion Register

This register enables inverting the output level from a pin. It is effective when the pin is in output mode regardless of port output mode or alternative output mode.

Access: This register can be read/written in 32-bit units.

Address: <PORT_base> + 0030_H + n × 40_H⁺¹

Value after reset: 0000 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5) in Section 2.4.1.1, List of Port Registers.

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PINVn_ 15 | PINVn_ 14 | PINVn_ 13 | PINVn_ 12 | PINVn_ 11 | PINVn_ 10 | PINVn_ 9 | PINVn_ 8 | PINVn_ 7 | PINVn_ 6 | PINVn_ 5 | PINVn_ 4 | PINVn_ 3 | PINVn_ 2 | PINVn_ 1 | PINVn_ 0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.24 PINVn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | PINVn_[15:0] | Specifies whether the output level from a pin is inverted or not. 0: Pin output level is not inverted. 1: Pin output level is inverted. |

NOTE

For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers.**

2.3.4.4 Configuration of Electrical Characteristics

(1) PUn/JPU0 — Pull-up Option Register

This register specifies whether an on-chip pull-up resistor is connected to an input pin.

Access: PUn: This register can be read/written in 16-bit units.
JPU0: This register can be read/written in 8-bit units.

Address: PUn: <PORT_base> + 400C_H + n × 40_H
JPU0: <JPORT0_base> + 400C_H*1

Value after reset: PUn: 0000_H
JPU0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | PUn_15 | PUn_14 | PUn_13 | PUn_12 | PUn_11 | PUn_10 | PUn_9 | PUn_8 | PUn_7 | PUn_6 | PUn_5 | PUn_4 | PUn_3 | PUn_2 | PUn_1 | PUn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.25 PUn Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 15 to 0 | PUn_[15:0] | Specifies whether an on-chip pull-up resistor is connected to the corresponding pin: 0: No on-chip pull-up resistor is connected. 1: On-chip pull-up resistor is connected. |

NOTES

- Do not set PUn.PUn_m = 1 and PDn.PDn_m = 1 to a single pin.
- The on-chip pull-up resistor has no effect when the pin is operated in output mode.
- The control bits of the JTAG pull-up option register (JPU0) are JPU0_[7:0].
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(2) PDn/JPD0 — Pull-down Option Register

This register specifies whether an on-chip pull-down resistor is connected to an input pin.

Access: PDn: This register can be read/written in 16-bit units.
JPD0: This register can be read/written in 8-bit units.

Address: PDn: <PORT_base> + 4010_H + n × 40_H
JPD0: <JPORT0_base> + 4010_H*1

Value after reset: PDn: 0000_H
JPD0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | PDn_15 | PDn_14 | PDn_13 | PDn_12 | PDn_11 | PDn_10 | PDn_9 | PDn_8 | PDn_7 | PDn_6 | PDn_5 | PDn_4 | PDn_3 | PDn_2 | PDn_1 | PDn_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.26 PDn Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 15 to 0 | PDn_[15:0] | Specifies whether a on-chip pull-down resistor is connected to the corresponding pin: 0: No on-chip pull-down resistor is connected. 1: On-chip pull-down resistor is connected. |

NOTES

- Do not set PUn.PUn_m = 1 and PDn.PDn_m = 1 to a single pin.
- The on-chip pull-down resistor has no effect when the pin is operated in output mode.
- The control bits of the JTAG pull-down option register (JPD0) are JPD0_[7:0].
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(3) PODCn/JPODC0 — Port Open-drain Control Register

This register selects push-pull or open-drain as the output buffer function.

Access: This register can be read/written in 32-bit units.

Address: PODCn: <PORT_base> + 4014_H + n × 40_H
JPODC0: <JPORT0_base> + 4014_H*1

Value after reset: PODC0: 0000 0400_H, PODC1 to 5: 0000 0000_H
JPODC0: 0000 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PODC n_15 | PODC n_14 | PODC n_13 | PODC n_12 | PODC n_11 | PODC n_10 | PODC n_9 | PODC n_8 | PODC n_7 | PODC n_6 | PODC n_5 | PODC n_4 | PODC n_3 | PODC n_2 | PODC n_1 | PODC n_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | *1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. PODC0: 1
PODC1 to PODC5, JPODC0: 0

Table 2.27 PODCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | PODCn_[15:0] | Specify the output circuit characteristics of Pn_m (m = 0 to 15), together with the PODCEn_[15:0] bits. |
| | PODCEn_m | PODCn_m |
| | 0 | 0 |
| | 0 | 1 |
| | 1 | 0 |
| | 1 | 1 |
| | | Output Circuit Characteristics |
| | | Push-pull |
| | | N-ch open-drain |
| | | Push-pull |
| | | P-ch open-drain |
| | Specify the output circuit characteristics of JPN_m (m = 0 to 15). | |
| | JPODCn_m | Output Circuit Characteristics |
| | 0 | Push-pull |
| | 1 | N-ch open-drain |

NOTES

- The control bits for the JTAG port open-drain control register (JPODC0) are JPODC0_[31:0].
- For details on the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.

(4) PODCEn — Port Open-drain Control Expansion Register

This register selects push-pull or open-drain as the output buffer function.

Access: This register can be read/written in 32-bit units.

Address: <PORT_base> + 403C_H + n × 40_H^{*1}

Value after reset: 0000 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5) in Section 2.4.1.1, List of Port Registers.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PODCE _{n_15} | PODCE _{n_14} | PODCE _{n_13} | PODCE _{n_12} | PODCE _{n_11} | PODCE _{n_10} | PODCE _{n_9} | PODCE _{n_8} | PODCE _{n_7} | PODCE _{n_6} | PODCE _{n_5} | PODCE _{n_4} | PODCE _{n_3} | PODCE _{n_2} | PODCE _{n_1} | PODCE _{n_0} |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.28 PODCEn Register Contents

| Bit Position | Bit Name | Function | |
|--------------|---------------------------------|---|-----------------|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | |
| 15 to 0 | PODCE _n [15:0] | Specify the output circuit characteristics of pin m (m = 0 to 15), together with the PODCn_ _[15:0] bits. | |
| | PODCE _n _m | PODCn_ _m | |
| | 0 | 0 | Push-pull |
| | 0 | 1 | N-ch open-drain |
| | 1 | 0 | Push-pull |
| | 1 | 1 | P-ch open-drain |

(5) PDSCn/JPDSC0 — Port Drive Strength Control Register

This register specifies the output drive strength of a port pin. High- or low-speed mode (high or low drive strength) of an output buffer is selectable by using this function.

Access: This register can be read/written in 32-bit units.

Address: PDSCn: <PORT_base> + 4018_H + n × 40_H
JPDSC0: <JPORT0_base> + 4018_H^{*1}

Value after reset: PDSC0: 0000 0400_H, PDSC1 to 5: 0000 0000_H
JPDSC0: 0000 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PDSCn _15 | PDSCn _14 | PDSCn _13 | PDSCn _12 | PDSCn _11 | PDSCn _10 | PDSCn _9 | PDSCn _8 | PDSCn _7 | PDSCn _6 | PDSCn _5 | PDSCn _4 | PDSCn _3 | PDSCn _2 | PDSCn _1 | PDSCn _0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | *1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. PDSC0: 1
PDSC1 to PDSC5, JPDSC0: 0

Table 2.29 PDSCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | PDSCn_[15:0] | Specifies the output buffer characteristics of pin m (m = 0 to 15), together with the PUCcn_[15:0] bits. |
| | PUCcn_m | PDSCn_m |
| | 0 | 0 |
| | 0 | 1 |
| | 1 | 0 |
| | 1 | 1 |
| | | Output Buffer Characteristics |
| | | SLOW |
| | | FAST |
| | | MIDDLE |
| | | MIDDLE |

NOTES

- The control bits for the JTAG port drive strength control register (JPDSC0) are JPDSC0_[31:0].
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.
- For the output buffer characteristics of the driving ability, see Section 37.9.2, Driving Ability.
- Settings for driving ability are effective even if an alternative function is in use on the pin.
- When the driving ability is set to “fast”, the pin produces a larger driving current than when the setting is “slow”. For the allowable total driving currents, see Section 37.8.3, Allowed Output Current.

(6) PUCcN/JPUCc0 — Port Universal Control Register

This register expands the function to specify the characteristics of an output buffer.

It can specify three output buffer characteristics, together with the Port Drive Strength Control Register (PDSCn).

Access: This register can be read/written in 32-bit units.

Address: PUCcN: <PORT_base> + 4028_H + n × 40_H
 JPUCc0: <JPORT0_base> + 4028_H*1

Value after reset: PUCcN: 0000 0000_H
 JPUCc0: 0000 0000_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PUCcN _15 | PUCcN _14 | PUCcN _13 | PUCcN _12 | PUCcN _11 | PUCcN _10 | PUCcN _9 | PUCcN _8 | PUCcN _7 | PUCcN _6 | PUCcN _5 | PUCcN _4 | PUCcN _3 | PUCcN _2 | PUCcN _1 | PUCcN _0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.30 PUCcN Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | PUCcN_[15:0] | Specifies the output buffer characteristics of pin m (m = 0 to 15), together with the PDSCn_[15:0] bits. |

| PUCcN_m | PDSCn_m | Output Buffer Characteristics |
|---------|---------|-------------------------------|
| 0 | 0 | SLOW |
| 0 | 1 | FAST |
| 1 | 0 | MIDDLE |
| 1 | 1 | MIDDLE |

NOTES

1. The control bits of the JTAG port universal control register (JPUCc0) are JPUCc0_[31:0].
2. For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.
3. For the output buffer characteristics of the driving ability, see Section 37.9.2, Driving Ability.
4. Settings for driving ability are effective even if an alternative function is in use on the pin.
5. When the driving ability is set to “fast”, the pin produces a larger driving current than when the setting is “slow”. For the allowable total driving currents, see Section 37.8.3, Allowed OutputCurrent.

(7) PISAn/JPISA0 — Port Input Buffer Selection Register

This register specifies the input buffer characteristics.

Access: PISAn: This register can be read/written in 16-bit units.
JPISA0: This register can be read/written in 8-bit units.

Address: PISAn: <PORT_base> + 402C_H + n × 40_H
JPISA0: <JPORT0_base> + 402C_H*1

Value after reset: PISA0: 0400_H, PISA1 to 4: 0000_H
JPISA0: 00_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | PISA n_15 | PISA n_14 | PISA n_13 | PISA n_12 | PISA n_11 | PISA n_10 | PISA n_9 | PISA n_8 | PISA n_7 | PISA n_6 | PISA n_5 | PISA n_4 | PISA n_3 | PISA n_2 | PISA n_1 | PISA n_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | *1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. PISA0: 1
PISA1 to PISA4, JPISA0: 0

Table 2.31 PISAn Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 15 to 0 | PISAn_[15:0] | Specify the input buffer characteristics. 0: Type 1 (Schmitt) 1: Type 2 (CMOS) |

NOTES

- For details about the definition of type 1 and type 2, see Section 2.3.5.4, Input Buffer Control (PISA).
- For details about the relevant port groups and bits, see Section 2.4.1.1, List of Port Registers.
- The control bits for the JTAG port input buffer select register (JPISA0) are JPISA0_[7:0]

2.3.4.5 Pin-unit Register

(1) PCRn_m/JPCR0_m — Port Control Register

Each register of a port group can be accessed via this register and a PCR register can set some functions of a single pin. For example, setting bit 6 of the PCRn_m register to 1 sets bit m of the PMCN register to 1 also.

Access: This register can be read/written in 32-bit units.

Address: PCRn_m: <PORT_base> + 2000_H + n × 40_H + 4_H × m
JPCR0_m: <JPORT0_base> + 2000_H + 4_H × m^{*1}

Value after reset: PCR0_10: 1140 0000_H, PCRn_m except for PCR0_10: 0000 0010_H
JPCR0_m: 0000 0010_H

Note 1. The number of valid bit positions (the value m) depends on the number of pins included in each device. See the list of registers by group (Table 2.36, List of Registers in Port Group 0, Table 2.37, List of Registers in Port Group 1, Table 2.38, List of Registers in Port Group 2, Table 2.39, List of Registers in Port Group 3, Table 2.40, List of Registers in Port Group 4, Table 2.41, List of Registers in Port Group 5, Table 2.42, List of Registers in Port Group JP0) in Section 2.4.1.1, List of Port Registers.

Note 2. The PINVn, PODCn, PODCEn, PUCn, and PDSCn registers cannot be set via this register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|------|----|------|-------|----|-----|------|----|------|------|-----|-----|-----|------|------|
| | — | PINV | — | PODC | PODCE | — | PUC | PDSC | — | PISA | — | — | PU | PD | PBDC | PIBC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | P | — | — | — | PPR | — | PMC | PIPC | PM | — | PFC | PFCE | PFC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *2 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R | R | R | R | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Note 1. PCR0_10: 1
PCRn_m except PCR0_10, JPCR0_m: 0

Note 2. PCR0_10: 0
PCRn_m except PCR0_10, JPCR0_m: 1

Table 2.32 PCRn_m Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 30 | PINV | The value of bit m of the PINVn register can be read. |
| 29 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 28 | PODC | The value of bit m of the PODCn register can be read. |
| 27 | PODCE | The value of bit m of the PODCEn register can be read. |
| 26 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 25 | PUC | The value of bit m of the PUCn register can be read. |
| 24 | PDSC | The value of bit m of the PDSCn register can be read. |
| 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 | PISA | Same function as bit m of the PISAn register. |
| 21, 20 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 19 | PU | Same function as bit m of the PUn register |

Table 2.32 PCRn_m Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 18 | PD | Same function as bit m of the PDn register |
| 17 | PBDC | Same function as bit m of the PBDCn register |
| 16 | PIBC | Same function as bit m of the PIBCn register |
| 15 to 13 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 12 | P | Same function as bit m of the Pn register |
| 11 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | PPR | The value of bit m of the PPRn register can be read. |
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | PMC | Same function as bit m of the PMcN register |
| 5 | PIPC | Same function as bit m of the PIPcN register |
| 4 | PM | Same function as bit m of the PMn register |
| 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | PFCAE | Same function as bit m of the PFCAEn register |
| 1 | PFCE | Same function as bit m of the PFCEn register |
| 0 | PFC | Same function as bit m of the PFCn register |

NOTE

For details about the relevant port groups and bits, see **Section 2.4.1.1, List of Port Registers**.

2.3.4.6 Example of Port Configuration Flowchart

The followings are examples of the port configuration flow.

CAUTION

When a pin is set to an alternative output mode while the PIPn.PIPn_m bit is 0, it may be temporarily switched to an alternative input mode. This occurs during the period between setting of the PMn.PMn_m bit to 1 and setting of the PIPn.PIPn_m bit to 0. Due to this possibility of a pin being temporarily switched to an alternative input mode, if a pin has an interrupt-related signal as an alternative function, disable the interrupt or make sure that it is ignored.

(1) Batch Setting

The following shows an example to collectively set a port group.

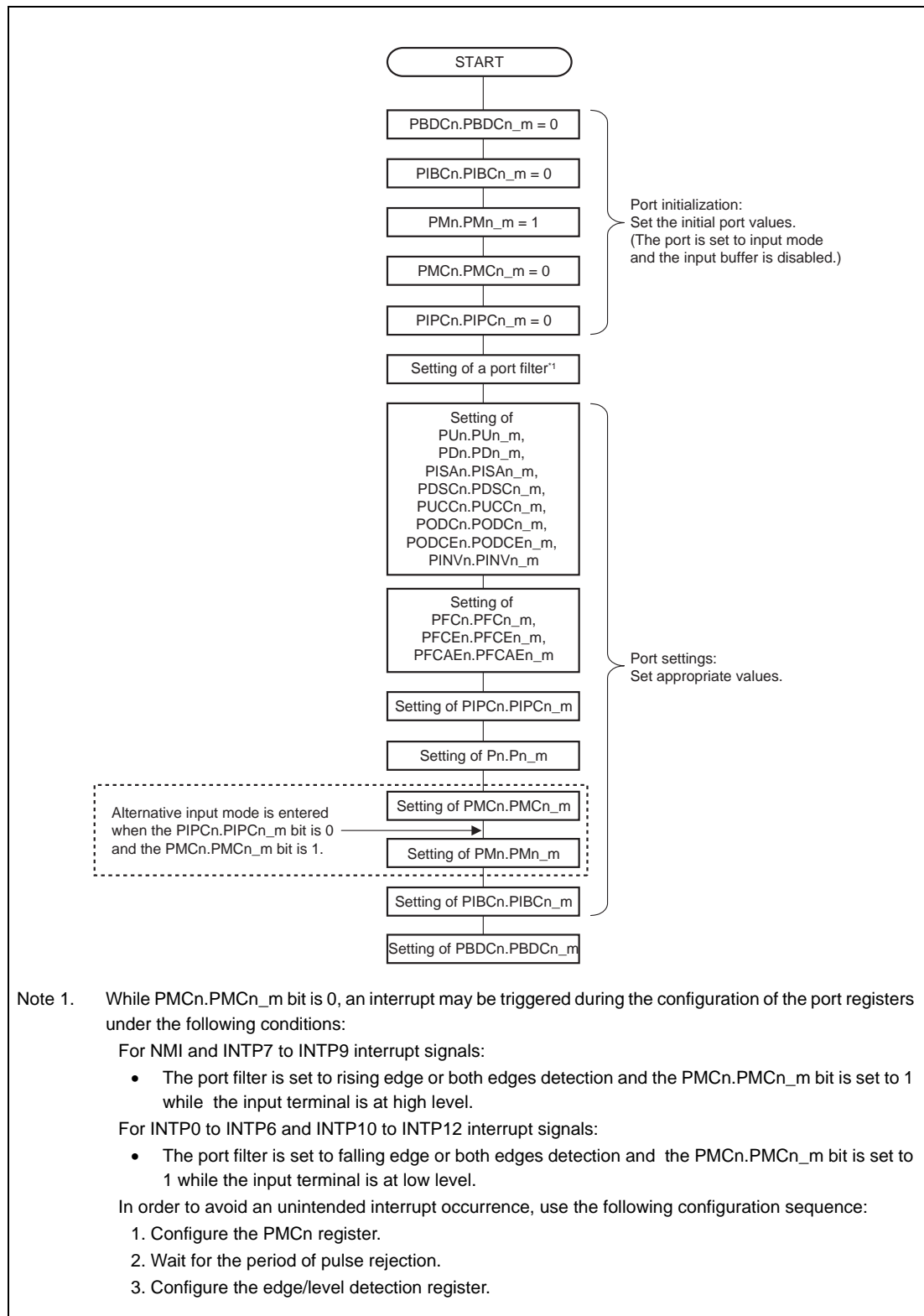


Figure 2.5 Example of Port Configuration Flow (batch setting)

(2) Individual Setting

The following shows an example to individually set a port group.

And it is capable of setting simultaneous bits at once within the configurable range described below by using PCRn_m register.

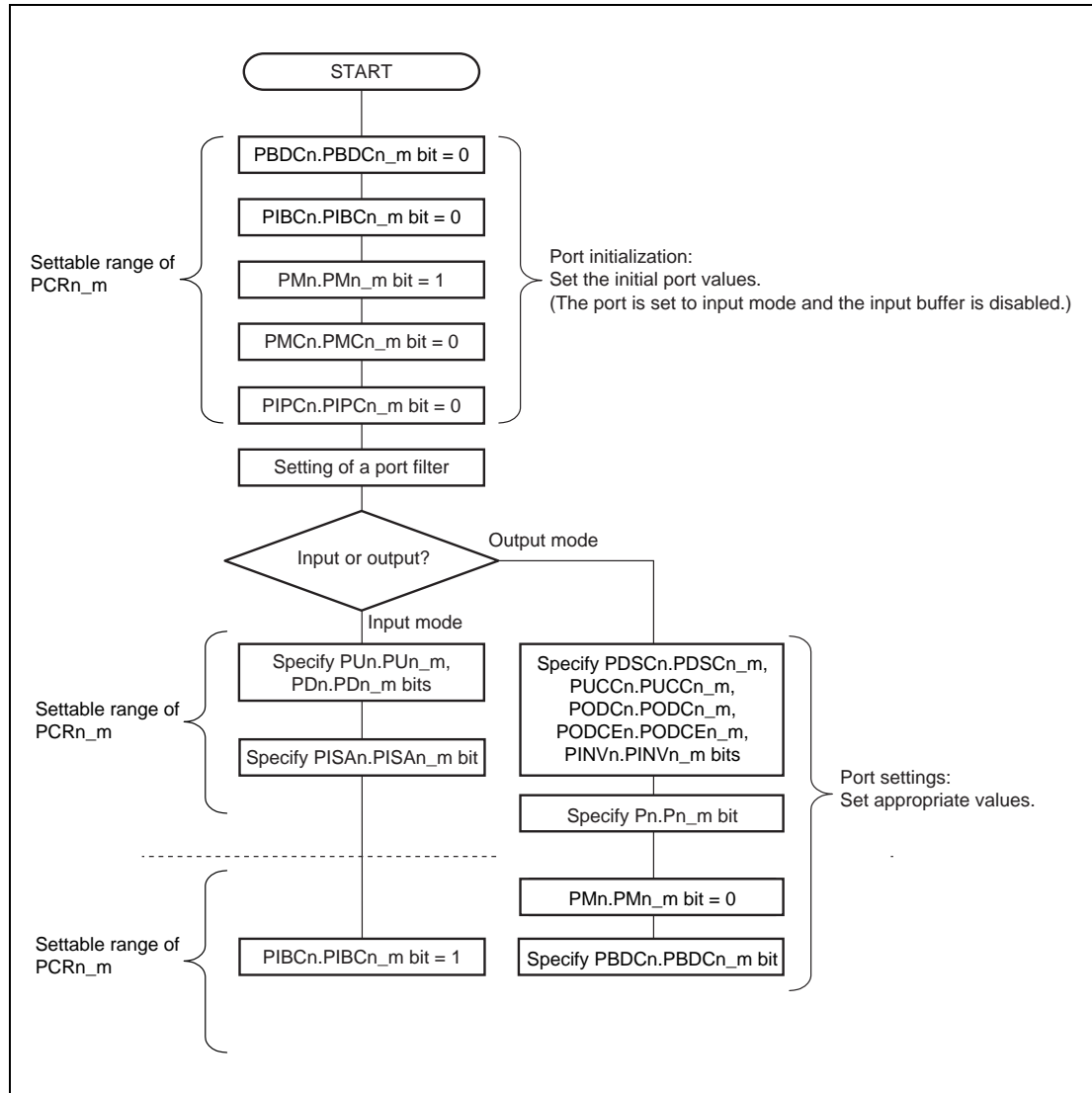


Figure 2.6 Example of Port Configuration Flow (in port mode)

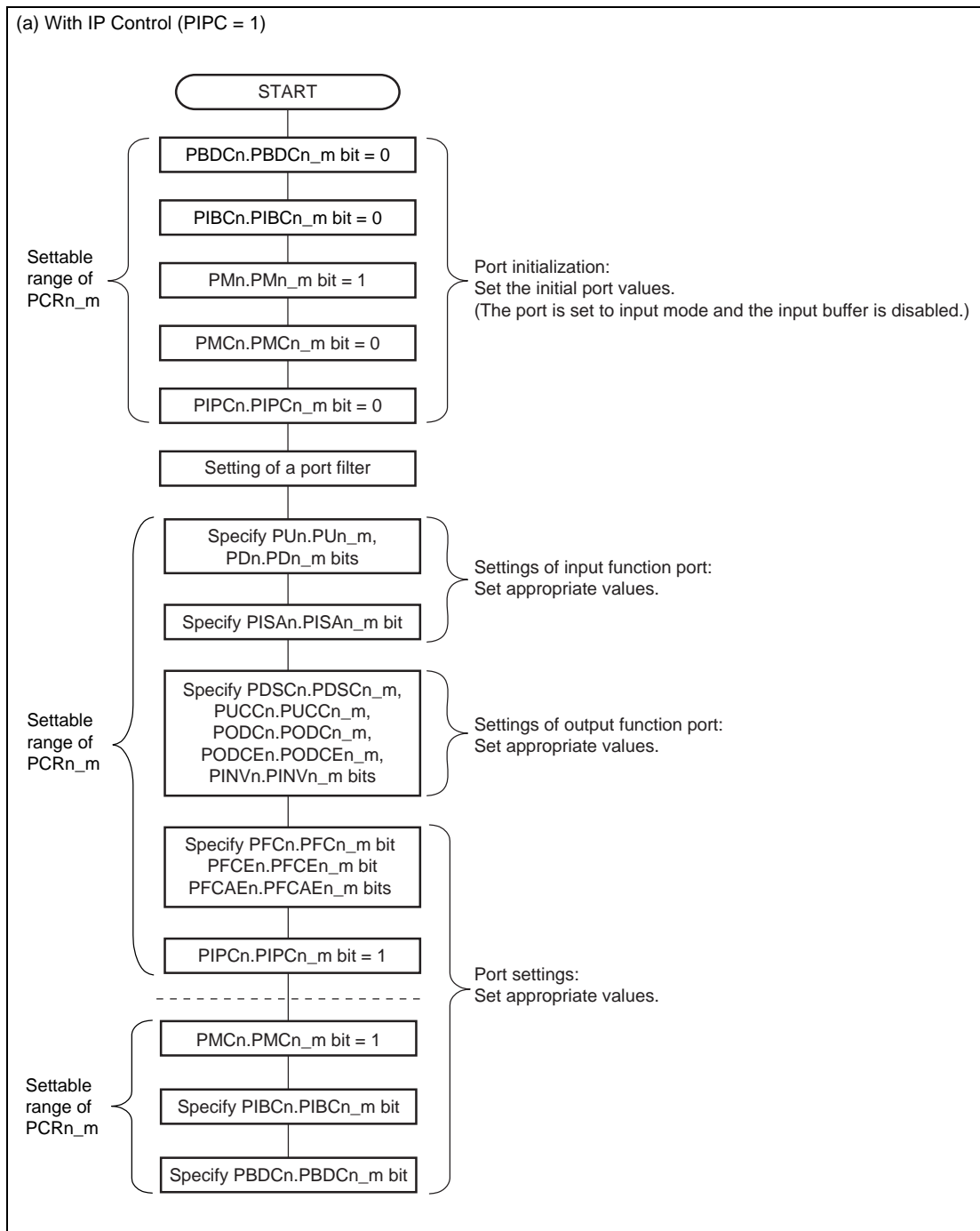


Figure 2.7 Example of Port Configuration Flow (in alternative mode)(1/2)

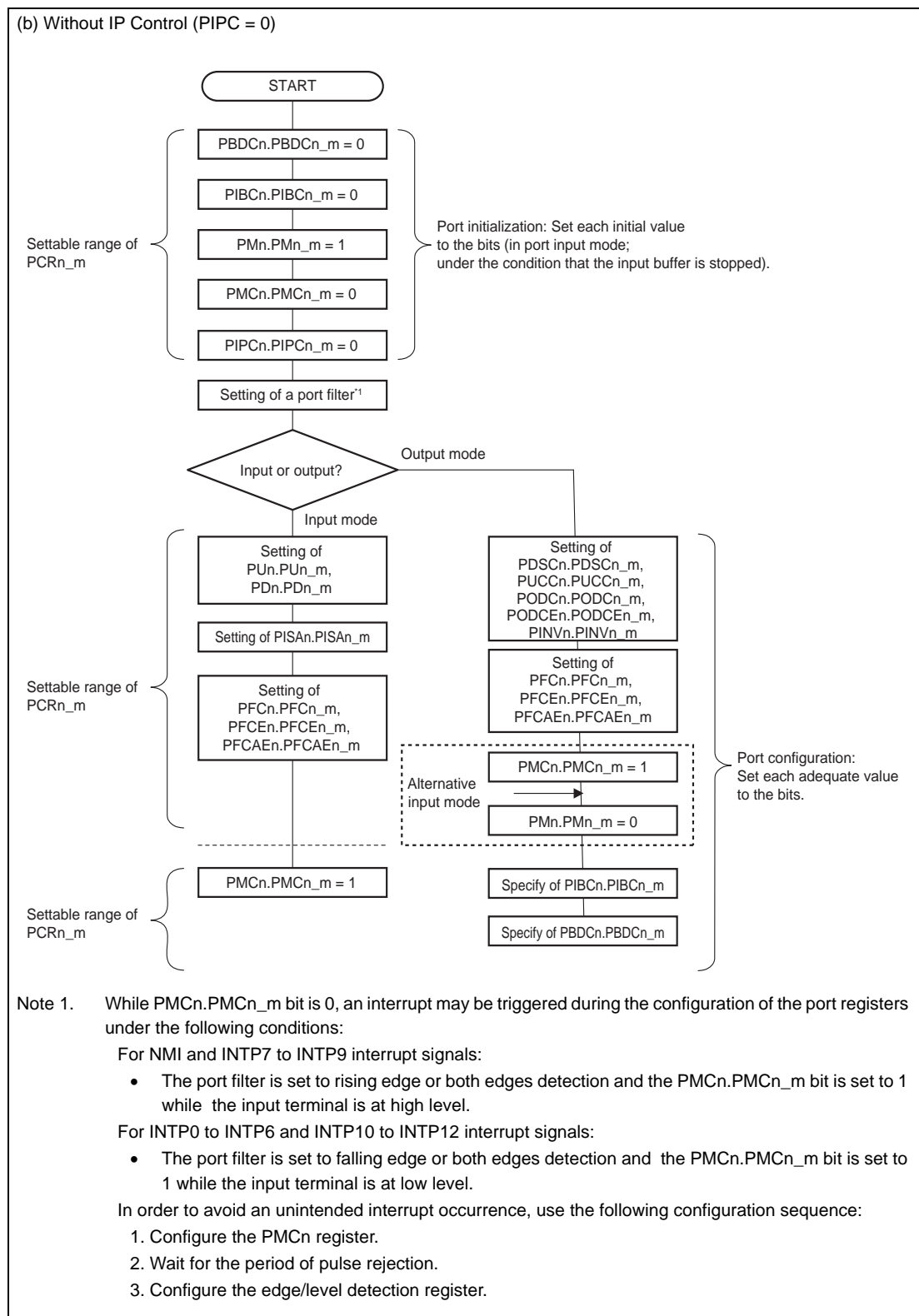


Figure 2.8 Example of Port Configuration Flow (in alternative mode) (2/2)

2.3.5 Functional Selection

2.3.5.1 Register Configuration in Use of the Alternative Function

When the pin alternative function is used, set $PMc_n_m = 1$ and select the alternative numbers of $PFCn_m$, $PFCEn_m$, and $PFCAEn_m$. In several peripheral functions, a single alternative I/O function is allocated to multiple pins. However, such an alternative function should not be enabled in multiple pins at the same time. For example, an external interrupt input, $INTP0$ is allocated to $P0_2$ and $P2_5$, but this alternative function can only be selected in either of them.

2.3.5.2 Alternative Function to be used in Direct I/O Control Alternative Mode

When the pins listed in **Table 2.33** are used, switch to direct I/O control alternative mode. When setting $PIPCn_m = 1$, the PMn_m value which has been set is ignored because the peripheral function enables or disables inputs and outputs of the buffer.

Table 2.33 List of the Pins which is Available for Direct I/O Control Alternative Mode Setting

| Function | Pin Name |
|----------|----------|
| CSIH0 | CSIH0SCO |
| | CSIH0SO |
| CSIH1 | CSIH1SCO |
| | CSIH1SO |
| CSIH2 | CSIH2SCO |
| | CSIH2SO |
| CSIH3 | CSIH3SCO |
| | CSIH3SO |
| CSIG0 | CSIG0SCO |
| | CSIG0SO |
| TSG30 | TSG30O1 |
| | TSG30O2 |
| | TSG30O3 |
| | TSG30O4 |
| | TSG30O5 |
| | TSG30O6 |
| TSG31 | TSG31O1 |
| | TSG31O2 |
| | TSG31O3 |
| | TSG31O4 |
| | TSG31O5 |
| | TSG31O6 |

CAUTION

Do not set $PIPCn_m$ to 1 at pins not listed in Table 2.33.

2.3.5.3 Register Setting in Use of an Analog Input Pin

Since ADCGnIm (SAR-A/D) for an analog input are always connected to the A/D converter, setting of a port register to select a pin function is not required.

2.3.5.4 Input Buffer Control (PISA)

This device can select the input buffer characteristics (type 1 or type 2) of a port with the PISAn register. The following table lists applicable pins.

Table 2.34 Selection of Input Buffer Characteristics (1/2)

| Port Name | Input Buffer Selection | | Remarks | Device | | | |
|-----------|------------------------|----------------------|---------|---------------|---------------|---------------|---------------|
| | Type 1 (PISAn_m = 0) | Type 2 (PISAn_m = 1) | | 100-pin (eVR) | 100-pin (DPS) | 144-pin (eVR) | 144-pin (DPS) |
| P0_3 | SHMT | CMOS | | — | — | √ | √ |
| P0_4 | SHMT | CMOS | | — | — | √ | √ |
| P0_5 | SHMT | CMOS | | — | — | √ | √ |
| P0_6 | SHMT | CMOS | | — | — | √ | √ |
| P0_7 | SHMT | CMOS | | — | — | √ | √ |
| P0_8 | SHMT | CMOS | | — | — | √ | √ |
| P0_9 | SHMT | CMOS | | — | — | √ | √ |
| P0_13 | SHMT | CMOS | | √ | √ | √ | √ |
| P1_1 | SHMT | CMOS | | √ | √ | √ | √ |
| P1_2 | SHMT | CMOS | | √ | √ | √ | √ |
| P1_3 | SHMT | CMOS | | √ | √ | √ | √ |
| P1_4 | SHMT | CMOS | | √ | √ | √ | √ |
| P2_0 | SHMT | CMOS | | √ | √ | √ | √ |
| P2_1 | SHMT | CMOS | | √ | √ | √ | √ |
| P2_2 | SHMT | CMOS | | √ | √ | √ | √ |
| P2_5 | SHMT | CMOS | | √ | √ | √ | √ |
| P2_7 | SHMT | CMOS | | √ | √ | √ | √ |
| P2_12 | SHMT | CMOS | | — | — | √ | √ |
| P3_0 | SHMT | CMOS | | — | — | √ | √ |
| P3_1 | SHMT | CMOS | | — | — | √ | √ |
| P3_14 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_0 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_1 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_2 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_3 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_4 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_5 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_6 | SHMT | CMOS | | √ | √ | √ | √ |
| P4_9 | SHMT | CMOS | | — | — | √ | √ |
| P4_10 | SHMT | CMOS | | — | — | √ | √ |
| P4_12 | SHMT | CMOS | | — | — | √ | √ |
| P4_13 | SHMT | CMOS | | — | — | √ | √ |
| JP0_0 | SHMT | CMOS | | √ | √ | √ | √ |
| JP0_1 | SHMT | CMOS | | √ | √ | √ | √ |

Table 2.34 Selection of Input Buffer Characteristics (2/2)

| Port Name | Input Buffer Selection | | Remarks | Device | | | |
|-----------|------------------------|----------------------|---------|---------------|---------------|---------------|---------------|
| | Type 1 (PISAn_m = 0) | Type 2 (PISAn_m = 1) | | 100-pin (eVR) | 100-pin (DPS) | 144-pin (eVR) | 144-pin (DPS) |
| JP0_2 | SHMT | CMOS | | √ | √ | √ | √ |
| JP0_3 | SHMT | CMOS | | √ | √ | √ | √ |
| JP0_5 | SHMT | CMOS | | √ | √ | √ | √ |

2.3.5.5 Output Buffer Control (PDSC, PUCC)

This device can select the output driver strength (SLOW, MIDDLE, or FAST) of a port with the PDSCn and PUCCn registers. The following table lists applicable pins.

Table 2.35 Selection of Output Driving Ability (1/3)

| Port Name | Device | | | |
|-----------|---------------|---------------|---------------|---------------|
| | 100-pin (eVR) | 100-pin (DPS) | 144-pin (eVR) | 144-pin (DPS) |
| P0_0 | √ | √ | √ | √ |
| P0_1 | √ | √ | √ | √ |
| P0_2 | √ | √ | √ | √ |
| P0_3 | — | — | √ | √ |
| P0_4 | — | — | √ | √ |
| P0_5 | — | — | √ | √ |
| P0_6 | — | — | √ | √ |
| P0_7 | — | — | √ | √ |
| P0_8 | — | — | √ | √ |
| P0_9 | — | — | √ | √ |
| P0_10 | √ | √ | √ | √ |
| P0_11 | — | — | √ | √ |
| P0_12 | — | — | √ | √ |
| P0_13 | √ | √ | √ | √ |
| P0_14 | — | — | √ | √ |
| P1_0 | — | — | √ | √ |
| P1_1 | √ | √ | √ | √ |
| P1_2 | √ | √ | √ | √ |
| P1_3 | √ | √ | √ | √ |
| P1_4 | √ | √ | √ | √ |
| P2_0 | √ | √ | √ | √ |
| P2_1 | √ | √ | √ | √ |
| P2_2 | √ | √ | √ | √ |
| P2_3 | √ | √ | √ | √ |
| P2_4 | √ | √ | √ | √ |
| P2_5 | √ | √ | √ | √ |
| P2_6 | √ | √ | √ | √ |
| P2_7 | √ | √ | √ | √ |
| P2_8 | √ | √ | √ | √ |
| P2_9 | √ | √ | √ | √ |
| P2_10 | — | — | √ | √ |
| P2_11 | — | — | √ | √ |
| P2_12 | — | — | √ | √ |
| P2_13 | — | — | √ | √ |
| P2_14 | — | — | √ | √ |
| P2_15 | — | — | √ | √ |
| P3_0 | — | — | √ | √ |
| P3_1 | — | — | √ | √ |
| P3_2 | — | — | √ | √ |

Table 2.35 Selection of Output Driving Ability (2/3)

| Port Name | Device | | | |
|-----------|---------------|---------------|---------------|---------------|
| | 100-pin (eVR) | 100-pin (DPS) | 144-pin (eVR) | 144-pin (DPS) |
| P3_3 | √ | √ | √ | √ |
| P3_4 | √ | √ | √ | √ |
| P3_5 | √ | √ | √ | √ |
| P3_6 | √ | √ | √ | √ |
| P3_7 | √ | √ | √ | √ |
| P3_8 | √ | √ | √ | √ |
| P3_9 | √ | √ | √ | √ |
| P3_10 | √ | √ | √ | √ |
| P3_11 | √ | √ | √ | √ |
| P3_12 | √ | √ | √ | √ |
| P3_13 | √ | √ | √ | √ |
| P3_14 | √ | √ | √ | √ |
| P4_0 | √ | √ | √ | √ |
| P4_1 | √ | √ | √ | √ |
| P4_2 | √ | √ | √ | √ |
| P4_3 | √ | √ | √ | √ |
| P4_4 | √ | √ | √ | √ |
| P4_5 | √ | √ | √ | √ |
| P4_6 | √ | √ | √ | √ |
| P4_7 | — | — | √ | √ |
| P4_8 | — | — | √ | √ |
| P4_9 | — | — | √ | √ |
| P4_10 | — | — | √ | √ |
| P4_11 | — | — | √ | √ |
| P4_12 | — | — | √ | √ |
| P4_13 | — | — | √ | √ |
| P4_14 | — | — | √ | √ |
| P5_0 | √ | √ | √ | √ |
| P5_1 | √ | √ | √ | √ |
| P5_2 | — | — | √ | √ |
| P5_3 | — | — | √ | √ |
| P5_4 | √ | √ | √ | √ |
| P5_5 | √ | √ | √ | √ |
| P5_6 | √ | √ | √ | √ |
| P5_7 | — | — | √ | √ |
| P5_8 | — | — | √ | √ |
| P5_9 | √ | √ | √ | √ |
| P5_10 | √ | √ | √ | √ |
| P5_11 | — | — | √ | √ |
| P5_12 | — | — | √ | √ |
| P5_13 | — | — | √ | √ |
| P5_14 | √ | √ | √ | √ |
| P5_15 | — | — | √ | √ |

Table 2.35 Selection of Output Driving Ability (3/3)

| Port Name | Device | | | |
|-----------|---------------|---------------|---------------|---------------|
| | 100-pin (eVR) | 100-pin (DPS) | 144-pin (eVR) | 144-pin (DPS) |
| JP0_0 | √ | √ | √ | √ |
| JP0_1 | √ | √ | √ | √ |
| JP0_3 | √ | √ | √ | √ |
| JP0_5 | √ | √ | √ | √ |

Table 2.36 List of Registers in Port Group 0 (2/2)

| Port Group Name | Register Name | R/W | Value after reset | Access Size | Bitmap | | | | | | | | | | | | | | | | Remarks |
|-----------------|---------------|-----|------------------------|-------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---------------|
| | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | PCR0_5 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_6 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_7 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_8 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_9 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_10 | R/W | 1140 0000 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | — | — | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_11 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | — | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_12 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | — | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_13 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR0_14 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | — | — | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |

Table 2.37 List of Registers in Port Group 1

| Port Group Name | Register Name | R/W | Value after reset | Access Size | Bitmap | | | | | | | | | | | | | | | | Remarks | | |
|-----------------|---------------|------------------------|------------------------|-------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---------|---------------|---------------|
| | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| 1 | P1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PSR1 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits | |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | | Upper 16 bits |
| | PPR1 | R | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PM1 | R/W | FFFF _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PMC1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PFC1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PFCE1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PNOT1 | W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PFCAE1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PMSR1 | R/W | 0000 FFFF _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ |
| | PMCSR1 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ |
| | PIBC1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PBDC1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PIPC1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | — | — | — | | | |
| | PU1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PD1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | | | |
| | PODC1 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | PDSC1 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | PUCC1 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | PISA1 | R/W | 0000 _H | 16 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | — | | | |
| | PODCE1 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | PINV1 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | PCR1_0 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ | √ | √ |
| PCR1_1 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | √ | √ | Lower 16 bits | |
| | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | √ | √ | √ | √ |
| PCR1_2 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | √ | √ | Lower 16 bits | |
| | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | √ | √ | √ | √ |
| PCR1_3 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | — | √ | √ | √ | √ | √ | √ | Lower 16 bits | |
| | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | √ | √ | √ | √ |
| PCR1_4 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | — | √ | √ | √ | √ | √ | √ | Lower 16 bits | |
| | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | √ | √ | √ | √ |

Table 2.38 List of Registers in Port Group 2 (2/2)

| Port Group Name | Register Name | R/W | Value after reset | Access Size | Bitmap | | | | | | | | | | | | | | | | Remarks |
|-----------------|---------------|-----|------------------------|-------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---------------|
| | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 2 | PCR2_10 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR2_11 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR2_12 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR2_13 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR2_14 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR2_15 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | Upper 16 bits |

Table 2.39 List of Registers in Port Group 3 (2/2)

| Port Group Name | Register Name | R/W | Value after reset | Access Size | Bitmap | | | | | | | | | | | | | | | | Remarks |
|-----------------|---------------|-----|------------------------|-------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---------------|
| | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 3 | PCR3_10 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | — | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR3_11 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR3_12 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR3_13 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR3_14 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | √ |

Table 2.40 List of Registers in Port Group 4 (2/2)

| Port Group Name | Register Name | R/W | Value after reset | Access Size | Bitmap | | | | | | | | | | | | | | | | Remarks |
|-----------------|---------------|-----|------------------------|-------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---------------|---------------|
| | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 4 | PCR4_10 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR4_11 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | √ | √ | √ | √ | Upper 16 bits | |
| | PCR4_12 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR4_13 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits |
| | PCR4_14 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | √ | √ | √ | √ | √ | Upper 16 bits |

Table 2.41 List of Registers in Port Group 5 (2/2)

| Port Group Name | Register Name | R/W | Value after reset | Access Size | Bitmap | | | | | | | | | | | | | | | | Remarks |
|-----------------|---------------|-----|------------------------|-------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---------------|
| | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 5 | PCR5_10 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR5_11 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR5_12 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR5_13 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR5_14 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | √ | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |
| | PCR5_15 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | √ | √ | — | √ | — | √ | Lower 16 bits |
| | | | | | — | √ | — | √ | √ | — | √ | √ | — | — | — | — | √ | √ | √ | √ | √ |

Table 2.42 List of Registers in Port Group JP0

| Port Group Name | Register Name | R/W | Value after reset | Access Size | Bitmap | | | | | | | | | | | | | | | | Remarks | | | |
|-----------------|---------------|------------------------|------------------------|-------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---------------|---------------|---------------|---------------|
| | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | |
| JP0 | JP0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | | | |
| | JPSR0 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | Lower 16 bits | | |
| | | | | | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | Upper 16 bits | | | |
| | JPPR0 | R | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | √ | √ | √ | | | |
| | JPM0 | R/W | FF _H | 8 | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | | | |
| | JPMC0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | | | |
| | JPFCE0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | | | |
| | JPNOT0 | W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | | |
| | JPMSR0 | R/W | 0000 00FF _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | Lower 16 bits | |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ |
| | JPMCSR0 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ | Lower 16 bits | |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | √ |
| | JPIBC0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | √ | √ | √ | | |
| | JPBDC0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | | |
| | JPU0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | √ | √ | √ | | |
| | JPD0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | √ | √ | √ | | |
| | JPODC0 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | — | √ | √ | Lower 16 bits |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | JPDSC0 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | Lower 16 bits | |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | JPUCC0 | R/W | 0000 0000 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | Lower 16 bits | |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | JPISA0 | R/W | 00 _H | 8 | — | — | — | — | — | — | — | — | — | — | — | — | √ | — | √ | √ | √ | √ | | |
| | JPCR0_0 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | — | — | √ | — | Lower 16 bits | | |
| | | | | | — | — | — | √ | — | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits | | | |
| | JPCR0_1 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | √ | — | √ | — | — | — | √ | — | Lower 16 bits | | |
| | | | | | — | — | — | √ | — | — | √ | √ | — | √ | — | — | √ | √ | √ | √ | Upper 16 bits | | | |
| | JPCR0_2 | R/W | 0000 0010 _H | 32 | — | — | — | — | — | — | — | √ | — | √ | — | — | — | — | — | √ | — | Lower 16 bits | | |
| | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | Upper 16 bits |
| | JPCR0_3 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | — | — | — | — | — | — | — | — | Lower 16 bits | | |
| | | | | | — | — | — | √ | — | — | √ | √ | — | √ | — | — | — | — | — | — | — | — | — | Upper 16 bits |
| | JPCR0_4 | R/W | 0000 0010 _H | 32 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | Lower 16 bits | | |
| — | | | | | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | Upper 16 bits | |
| JPCR0_5 | R/W | 0000 0010 _H | 32 | — | — | — | √ | — | — | — | √ | — | — | — | — | — | — | — | — | — | Lower 16 bits | | | |
| | | | | — | — | — | √ | — | — | √ | √ | — | √ | — | — | — | — | — | — | — | — | — | Upper 16 bits | |

2.4.1.2 List of Alternative Function Pins

Table 2.43 to Table 2.49 show the list of alternative functions of each port pin. In the tables, “—” means a reserved bit which cannot be selected.

2.4.1.3 Port 0 (P0)

Table 2.43 Port 0 (P0)

| Port Mode (PMCO_m = 0) | Alternative Mode (PMCO_m = 1) | | | | | | | | | | | | | | Dedicated Function | | | Device | | |
|------------------------|-------------------------------|-----------|--------------------|-----------|-----------------|-----------|-----------------|------------------------|-----------------|--------------------|-----------------|--------|---------|-----|--------------------|-----|-----|--------|-----|--|
| | 1st Alternative | | 2nd Alternative | | 3rd Alternative | | 4th Alternative | | 5th Alternative | | 6th Alternative | | 100-pin | | 144-pin | | eVR | DPS | DPS | |
| | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | eVR | DPS | eVR | DPS | | | | |
| P0_0 | TAUJ01 | TAUJ001 | TAUD013 | SENT3SPCO | TAUD1O14 | TAUD1O15 | ADCG0CNV2 | TSG31CLKI | PSI50DOUT | RLIN30TX | | | √ | √ | √ | √ | | | | |
| P0_1 | TAUJ02 | TAUJ002 | SENT4RXC | SENT4SPCO | | | ADCG0CNV3 | TAUD0O10 | TAUD0O10 | RLIN31RX/ INTP4 | | | √ | √ | √ | √ | | | | |
| P0_2 | SCI30RX/ INTP0 | | TAPA1ESO/ INTP8 | | TAUD210 | ADCG0CNV0 | TAUD2O0 | SCI32RX/ INTP2 | TSG31O0 | SENT5RX | SENT5SPCO | | √ | √ | √ | √ | | | | |
| P0_3 | | SCI30TX | | TAUD1O8 | TAUD211 | ADCG0CNV1 | TAUD2O1 | TSG31O1 | TSG31O1 | SENT5SPCO | | | — | — | — | — | — | — | — | |
| P0_4 | SCI30SCI | SCI30SCO | | TAUD1O9 | TAUD212 | ADCG0CNV2 | TAUD2O2 | TSG31O2 | TSG31O2 | SENT4SPCO | | | — | — | — | — | — | — | — | |
| P0_5 | TAUJ10 | TAUJ100 | | TAUD1O10 | TAUD213 | ADCG0CNV3 | TAUD2O3 | TSG31O3 | TSG31O3 | SENT3SPCO | | | — | — | — | — | — | — | — | |
| P0_6 | TAUJ11 | TAUJ101 | | TAUD1O11 | TAUD214 | | TAUD2O4 | TSG31O4 | TSG31O4 | SENT2SPCO | | | — | — | — | — | — | — | — | |
| P0_7 | TAUJ12 | TAUJ102 | | TAUD1O12 | TAUD215 | | TAUD2O5 | TSG31O5 | TSG31O5 | SENT1SPCO | | | — | — | — | — | — | — | — | |
| P0_8 | TAUJ13 | TAUJ103 | | TAUD1O13 | TAUD216 | | TAUD2O6 | TSG31O6 | TSG31O6 | SENT0SPCO | | | — | — | — | — | — | — | — | |
| P0_9 | | ADCG1CNV1 | | TAUD1O14 | TAUD217 | ADCG0CNV4 | TAUD2O7 | TSG31O7 | TSG31O7 | | | | — | — | — | — | — | — | — | |
| P0_10 | | | | ADCG1CNV0 | | | | | | | | | √ | √ | √ | √ | √ | √ | √ | |
| P0_11 | | | | ADCG1CNV2 | | | | TSG31PTS10/ ENCA0E0 | | | | | — | — | — | — | — | — | — | |
| P0_12 | | | | ADCG1CNV3 | | | | TSG31PTS11/ ENCA0E1 | | | | | — | — | — | — | — | — | — | |
| P0_13 | | OSTM00 | | TAUD1O15 | TAUD217 | | TAUD2O7 | TSG31PTS12/ ENCA0E2 | | SCI31SCI | SCI31SCO | | √ | √ | √ | √ | √ | √ | √ | |
| P0_14 | | | | ADCG1CNV4 | | | | | | | | | — | — | — | — | — | — | — | |

Note: P0_10: RESETOUT

P0_10 is in initial state, RESETOUT, and outputs low level.

After reset release, the setting will be as below.

- PM0.PM0_10 = 0: output port
- PODC0.PODC0_10: Nch open-drain output

After reset release, RESETOUT output is stopped by changing the setting above.

When a reset is generated during operation, this pin becomes RESETOUT and outputs low level.

For the state of pins, see Section 8.4.8, Reset Output (RESETOUT).

2.4.1.4 Port 1 (P1)

Table 2.44 Port 1 (P1)

| Port Mode (PMC1_m = 0) | Alternative Mode (PMC1_m = 1) | | | | | | | | | | | | | | | | | | Dedicated Function | | | | Device | | | |
|------------------------|-------------------------------|---------|-----------------|---------|-----------------|----------|--------------------|-----------|-----------------|--------|-----------------|--------|-----------------|--------|-----------------|--------|-----------------|--------|--------------------|-----|---------|-----|--------|--|--|--|
| | 1st Alternative | | 2nd Alternative | | 3rd Alternative | | 4th Alternative | | 5th Alternative | | 6th Alternative | | 7th Alternative | | 8th Alternative | | 9th Alternative | | 100-pin | | 144-pin | | | | | |
| | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | eVR | DPS | eVR | DPS | | | | |
| P1_0 | | | | | | | | | | | | | | | | | | | — | — | √ | √ | | | | |
| P1_1 | TAUJ20 | TAUJ200 | TAUJ10 | TAUJ100 | TAUD212 | TAUD2012 | RLIN30RX/ INTP3 | CSIH2CSS0 | TAUD016 | | CSIH3RY1 | | | | | | | | √ | √ | √ | √ | | | | |
| P1_2 | TAUJ21 | TAUJ201 | TAUJ11 | TAUJ101 | TAUD213 | TAUD2013 | CSIH2SI | | TAUD018 | | CSIH3SI | | | | | | | | √ | √ | √ | √ | | | | |
| P1_3 | TAUJ22 | TAUJ202 | TAUJ12 | TAUJ102 | TAUD214 | TAUD2014 | | CSIH2SO | TAUD010 | | | | | | | | | | √ | √ | √ | √ | | | | |
| P1_4 | TAUJ23 | TAUJ203 | TAUJ13 | TAUJ103 | TAUD215 | TAUD2015 | CSIH2SCI | CSIH2SCO | | | CSIH3SCI | | | | | | | | √ | √ | √ | √ | | | | |

2.4.1.5 Port 2 (P2)

Table 2.45 Port 2 (P2)

| Port Mode (PMC2_m = 0) | Alternative Mode (PMC2_m = 1) | | | | | | | | | | | | | | Dedicated Function | | Device | |
|------------------------|-------------------------------|-----------|-----------------|-----------|--------------------|----------|-----------------|----------|-----------------|----------|-----------------|-----------|-------------|-------------|--------------------|-------------|--------|--|
| | 1st Alternative | | 2nd Alternative | | 3rd Alternative | | 4th Alternative | | 5th Alternative | | 6th Alternative | | 100-pin eVR | 100-pin DPS | 144-pin eVR | 144-pin DPS | | |
| | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | | | | | | |
| P2_0 | RSCAN0RX0/ INTP5 | | CSIH2SI | | TAUD2011 | TAUD2011 | CSIH3CSS6 | | | | | | | ✓ | ✓ | ✓ | ✓ | |
| P2_1 | | RSCAN0TX0 | | CSIH2SO | TAUD2012 | TAUD2012 | CSIH3CSS7 | | TSG3007 | | | | | ✓ | ✓ | ✓ | ✓ | |
| P2_2 | RSCAN0RX1/ INTP6 | | CSIH2SCI | CSIH2SCO | TAUD2013 | TAUD2013 | TPBA00 | | TSG3000 | | | CSIH0CSS5 | | ✓ | ✓ | ✓ | ✓ | |
| P2_3 | | RSCAN0TX1 | CSIH2RYI | CSIH2CSS0 | TAUD2014 | TAUD2014 | TPBA10 | | TSG3001 | | | | | ✓ | ✓ | ✓ | ✓ | |
| P2_4 | | CSIH2RYO | | CSIH0CSS2 | RLIN31TX | TAUJ103 | CSIH0SI | | | TAUD2111 | TAUD2011 | | | ✓ | ✓ | ✓ | ✓ | |
| P2_5 | SCI30RX/ INTP0 | | | CSIH0CSS3 | RLIN31RX/ INTP4 | | CSIH3SO | | TSG3002 | | | CSIH0SO | | ✓ | ✓ | ✓ | ✓ | |
| P2_6 | | SCI30TX | | OSTM1O | CSIH0SCI | CSIH0SCO | CSIH3SI | | TSG3003 | TAUD110 | TAUD100 | | | ✓ | ✓ | ✓ | ✓ | |
| P2_7 | SCI30SCI | SCI30SCO | | CSIH0CSS5 | CSIH1SI | | CSIH3SCI | | TSG3004 | TAUD111 | TAUD101 | | | ✓ | ✓ | ✓ | ✓ | |
| P2_8 | SCI31RX/ INTP1 | CSIH3RYO | | CSIH0CSS6 | CSIH1SO | | CSIH3RYI | | TSG3005 | TAUD112 | TAUD102 | | | ✓ | ✓ | ✓ | ✓ | |
| P2_9 | | SCI31TX | | CSIH0CSS7 | CSIH1SCI | CSIH1SCO | CSIH3CSS1 | | TSG3006 | TAUD113 | TAUD103 | | | ✓ | ✓ | ✓ | ✓ | |
| P2_10 | SENT5RX | SENT5SPCO | | | TAUD2110 | TAUD2010 | | CSIH0RYI | CSIH0CSS6 | | OSTM1O | | | — | — | — | — | |
| P2_11 | TAUD10 | SENT5SPCO | | TAUD100 | TAUD111 | | TAUD101 | | CSIH0CSS5 | CSIH1RYI | CSIH0RYO | | | — | — | — | — | |
| P2_12 | TAUD12 | | | TAUD102 | TAUD113 | | TAUD103 | | CSIH0CSS4 | | CSIH0SO | | | — | — | — | — | |
| P2_13 | TAUD14 | | | TAUD104 | TAUD115 | | TAUD105 | | CSIH0CSS3 | | CSIH1RYO | | | — | — | — | — | |
| P2_14 | TAUD16 | | | TAUD106 | TAUD117 | | TAUD107 | | CSIH0CSS2 | | | | | — | — | — | — | |
| P2_15 | TAUD18 | | | TAUD108 | TAUD119 | | TAUD109 | CSIH0RYI | | | | | | — | — | — | — | |

2.4.1.6 Port 3 (P3)

Table 2.46 Port 3 (P3)

| Port Mode (PMC3 m = 0) | Alternative Mode (PMC3_m = 1) | | | | | | | | | | | | | | Dedicated Function | | Device | | | |
|------------------------|-------------------------------|-----------|-----------------|------------|-----------------|------------|-----------------|------------|-----------------|-----------|-----------------|------------|------------|------------|--------------------|------------|------------|------------|------------|---------|
| | 1st Alternative | | 2nd Alternative | | 3rd Alternative | | 4th Alternative | | 5th Alternative | | 6th Alternative | | 100-pin | | 144-pin | | eVR | DPS | | |
| | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | eVR | DPS | eVR | DPS | | | | |
| P3_0 | TAUD110 | TAUD1O10 | TAUD1111 | TAUD1O11 | TAUD1O11 | TAUD1O11 | TAUD1O11 | TAUD1O11 | CSH0SO | CSIG0SI | CSIG0SI | CSIG0SO | CSIG0SO | CSIG0SO | CSIG0SO | CSIG0SO | CSIG0SO | CSIG0SO | CSIG0SO | |
| P3_1 | TAUD112 | TAUD1O12 | TAUD1113 | TAUD1O13 | TAUD1O13 | TAUD1O13 | TAUD1O13 | TAUD1O13 | CSH0SCI | CSH0SCI | CSH0SCI | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | |
| P3_2 | TAUD114 | TAUD1O14 | TAUD1115 | TAUD1O15 | TAUD1O15 | TAUD1O15 | TAUD1O15 | TAUD1O15 | CSH0SCI | CSH0SCI | CSH0SCI | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | CSH0SCO | |
| P3_3 | CSIH0YI | CSIH2CSS1 | CSIH3CSS1 | CSIH3CSS1 | CSIH3CSS1 | CSIH3CSS1 | CSIH3CSS1 | CSIH3CSS1 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | TAUD100 | |
| P3_4 | SCI32RX/INTP2 | TAUD112 | RLIN30RX/INTP3 | CSIG0RYO*1 | CSIG0RYO*1 | CSIG0RYO*1 | CSIG0RYO*1 | CSIG0RYO*1 | CSIH3CSS3 | CSIH3CSS3 | CSIH3CSS3 | TAUD102 | TAUD102 | TAUD102 | TAUD102 | TAUD102 | TAUD102 | TAUD102 | TAUD102 | |
| P3_5 | SCI31SCI | TAUD000 | SCI31SCO | CSIH2CSS3 | CSIH2CSS3 | CSIH2CSS3 | CSIH2CSS3 | CSIH2CSS3 | CSIH2CSS3 | CSIH2CSS3 | CSIH2CSS3 | TAUD000 | TAUD000 | TAUD000 | TAUD000 | TAUD000 | TAUD000 | TAUD000 | TAUD000 | |
| P3_6 | CSIH0SI | TSG31O0 | TSG31O0 | TSG31O0 | TSG31O0 | TSG31O0 | TSG31O0 | TSG31O0 | CSIH3RYI | CSIH3RYI | CSIH3RYI | CSH2CSS4 | CSH2CSS4 | CSH2CSS4 | CSH2CSS4 | CSH2CSS4 | CSH2CSS4 | CSH2CSS4 | CSH2CSS4 | |
| P3_7 | CSIH0SO | TSG31O1 | RSCAN0RX0/INTP5 | INTP5 | INTP5 | INTP5 | INTP5 | INTP5 | CSIH3RYI | CSIH3RYI | CSIH3RYI | CSH2CSS5 | CSH2CSS5 | CSH2CSS5 | CSH2CSS5 | CSH2CSS5 | CSH2CSS5 | CSH2CSS5 | CSH2CSS5 | |
| P3_8 | CSIH0SCI | TSG31O2 | TSG31O2 | RSCAN0TX0 | RSCAN0TX0 | RSCAN0TX0 | RSCAN0TX0 | RSCAN0TX0 | CSIH0SSI | CSIH0SSI | CSIH0SSI | CSH2CSS6 | CSH2CSS6 | CSH2CSS6 | CSH2CSS6 | CSH2CSS6 | CSH2CSS6 | CSH2CSS6 | CSH2CSS6 | |
| P3_9 | SCI32SCI | TSG31O3 | TSG31O3 | CSIH3CSS5 | CSIH3CSS5 | CSIH3CSS5 | CSIH3CSS5 | CSIH3CSS5 | CSIH0RYI | CSIH0RYI | CSIH0RYI | CSH2CSS7 | CSH2CSS7 | CSH2CSS7 | CSH2CSS7 | CSH2CSS7 | CSH2CSS7 | CSH2CSS7 | CSH2CSS7 | |
| P3_10 | TAUJ03 | TSG31O4 | TSG31O4 | ADCG0CNV4 | ADCG0CNV4 | ADCG0CNV4 | ADCG0CNV4 | ADCG0CNV4 | TAUD116 | TAUD116 | TAUD116 | ERROROUT_C | ERROROUT_C | ERROROUT_C | ERROROUT_C | ERROROUT_C | ERROROUT_C | ERROROUT_C | ERROROUT_C | |
| P3_11 | TAUJ03 | SENT4SPCO | SENT4SPCO | CSIH3CSS5 | CSIH3CSS5 | CSIH3CSS5 | CSIH3CSS5 | CSIH3CSS5 | TAUD118 | TAUD118 | TAUD118 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | |
| P3_12 | CSIH1RYO | TSG31O5 | RSCAN0RX1/INTP6 | INTP6 | INTP6 | INTP6 | INTP6 | INTP6 | SCI30TX | SCI30TX | SCI30TX | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | CSH0CSS0 | |
| P3_13 | CSIH1RYI | TSG31O6 | TSG31O6 | RSCAN0TX1 | RSCAN0TX1 | RSCAN0TX1 | RSCAN0TX1 | RSCAN0TX1 | TAUD110 | TAUD110 | TAUD110 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | |
| P3_14 | CSIH1SI | TPBA10 | TPBA10 | EXTCLK00 | EXTCLK00 | EXTCLK00 | EXTCLK00 | EXTCLK00 | NMI | NMI | NMI | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | CSH0CSS1 | |
| | | | | | | | | | | | | | | | | | | | FLMD1 | TSG31O7 |

Note 1. Only in 144-pin products.

2.4.1.7 Port 4 (P4)

Table 2.47 Port 4 (P4)

| Port Mode (PMC4_m = 0) | Alternative Mode (PMC4_m = 1) | | | | | | | | | | | | Dedicated Function | | Device | |
|------------------------|-------------------------------|------------------------|-----------------|---------|-----------------|-----------|--------------------|--------------------|------------------|------------------------|-----------------|-----------|--------------------|---------|--------|-----|
| | 1st Alternative | | 2nd Alternative | | 3rd Alternative | | 4th Alternative | | 5th Alternative | | 6th Alternative | | 100-pin | 144-pin | DPS | DPS |
| | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | eVR | eVR | | |
| P4_0 | | CSH1SO | TAUJ00 | TAUJ00 | RLIN30RX/INTP3 | | TSG30PTS10/ENCA0E0 | | | FLXA0TXDA | | ADCG1CNV0 | ✓ | ✓ | ✓ | ✓ |
| P4_1 | CSH1SCI | CSH1SCO | TAUJ01 | TAUJ001 | | RLIN30TX | | TSG30PTS11/ENCA0E1 | | FLXA0TXEN _A | | ADCG1CNV1 | ✓ | ✓ | ✓ | ✓ |
| P4_2 | RSCAN0RX1/INTP6 | CSH1RYO | TAUJ02 | TAUJ002 | SCI30SCI | | TSG30PTS2/ENCA0EC | | FLXA0RXDA/INTP11 | | ADCG1CNV2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| P4_3 | CSH2RY1 | RSCAN0TX1 | TAUJ03 | TAUJ003 | | OSTM10 | | TSG30CLKI | CSH1CSS0 | FLXA0RXDB/INTP12 | TAUD210 | TAUD200 | ✓ | ✓ | ✓ | ✓ |
| P4_4 | CSH2SI | | TAPA1ESO/INTP8 | | | CSH2CSS7 | | TSG31PTS10/ENCA1E0 | CSH1CSS1 | FLXA0STPW _T | TAUD211 | TAUD201 | ✓ | ✓ | ✓ | ✓ |
| P4_5 | | CSH2SO | SCI30RX/INTP0 | | RSCAN0RX0/INTP5 | EXTCLK10 | | TSG31PTS11/ENCA1E1 | CSH1CSS2 | | TAUD212 | TAUD202 | ✓ | ✓ | ✓ | ✓ |
| P4_6 | CSH2SCI | CSH2SCO | | SCI30TX | | RSCAN0TX0 | | TSG31PTS12/ENCA1EC | CSH1CSS3 | | TAUD213 | TAUD203 | ✓ | ✓ | ✓ | ✓ |
| P4_7 | RSCAN0RX1/INTP6 | CSH2RYO | | TAUD100 | ENCA0TIN0 | CSH2CSS0 | | TSG31CLKI | | CSH1CSS4 | TAUD214 | TAUD204 | — | — | ✓ | ✓ |
| P4_8 | FLXA0RXDA/INTP11 | | | TAUD101 | ENCA0TIN1 | CSH2CSS1 | | CSH1SSI | | CSH1CSS5 | TAUD215 | TAUD205 | — | — | ✓ | ✓ |
| P4_9 | | FLXA0TXDA | | TAUD102 | ENCA1TIN0 | CSH2CSS2 | | CSH1RY1 | | CSH1CSS6 | TAUD216 | TAUD206 | — | — | ✓ | ✓ |
| P4_10 | | FLXA0TXEN _A | | TAUD103 | ENCA1TIN1 | CSH2CSS3 | | | CSH1RYO | CSH1CSS7 | TAUD217 | TAUD207 | — | — | ✓ | ✓ |
| P4_11 | FLXA0RXDB/INTP12 | | | TAUD104 | | CSH2CSS4 | | TAUJ110 | TAUJ100 | | TAUD218 | TAUD208 | — | — | ✓ | ✓ |
| P4_12 | | FLXA0TXDB | | TAUD105 | | CSH2CSS5 | | TAUJ111 | TAUJ101 | | TAUD219 | TAUD209 | — | — | ✓ | ✓ |
| P4_13 | | FLXA0TXEN _B | | TAUD106 | CSH2SSI | | | TAUJ112 | TAUJ102 | | TAUD2110 | TAUD2010 | — | — | ✓ | ✓ |
| P4_14 | FLXA0STPW _T | | | TAUD107 | | CSH2CSS6 | | TAUJ113 | TAUJ103 | | TAUD2111 | TAUD2011 | — | — | ✓ | ✓ |

2.4.1.8 Port 5 (P5)

Table 2.48 Port 5 (P5)

| Port Mode (PMC5_m = 0) | Alternative Mode (PMC5_m = 1) | | | | | | | | | | | | | | Dedicated Function | | Device | |
|------------------------|-------------------------------|-----------|-----------------|-----------|-----------------|-----------|-----------------|-----------|-----------------|-----------|--------------------|-----------|-------|--------|--------------------|-----|--------|-----|
| | 1st Alternative | | 2nd Alternative | | 3rd Alternative | | 4th Alternative | | 5th Alternative | | 6th Alternative | | | | | | | |
| | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | eVR | DPS | eVR | DPS |
| P5_0 | CSIG0SI | TAUD000 | TAUD001 | TAUD000 | TAUD001 | TAUD001 | TAUD001 | TAUD001 | TAUD001 | CSIH2CSS1 | SCI30RX/ INTP0 | | | | ✓ | ✓ | ✓ | ✓ |
| P5_1 | CSIG0SO | TAUD002 | TAUD003 | ADCG0CNV0 | TAUD003 | TAUD003 | TAUD003 | TAUD003 | TAUD003 | CSIH2CSS2 | TAUD011 | SCI30TX | | | ✓ | ✓ | ✓ | ✓ |
| P5_2 | CSIG0SCI | TAUD002 | TAUD014 | | TAUD003 | TAUD003 | TAUD003 | TAUD003 | TAUD015 | CSIH2CSS0 | SCI30SCI | SCI30SCO | | | — | — | — | — |
| P5_3 | | | TAUD017 | | | | | | TAUD016 | | | | | | — | — | — | — |
| P5_4 | CSIG0SCI | TAUD004 | TAUD015 | ADCG0CNV1 | TAUD005 | TAUD005 | TAUD005 | TAUD005 | TAUD014 | CSIH2CSS3 | SCI30SCI | SCI30SCO | | | ✓ | ✓ | ✓ | ✓ |
| P5_5 | SENT0RX | TAUD006 | TAUD017 | ADCG0CNV2 | TAUD007 | TAUD007 | TAUD007 | TAUD007 | TAUD016 | CSIH2CSS4 | SCI31RX/ INTP1 | RSCAN0TX2 | | | ✓ | ✓ | ✓ | ✓ |
| P5_6 | RSCAN0RX2 /INTP10 | TAUD008 | TAUD019 | ADCG0CNV3 | TAUD009 | TAUD009 | TAUD009 | TAUD009 | TAUD018 | CSIH2CSS5 | | SCI31TX | | | ✓ | ✓ | ✓ | ✓ |
| P5_7 | | TAUD010 | TAUD011 | ADCG0CNV4 | TAUD011 | TAUD011 | TAUD011 | TAUD011 | TAUD010 | CSIH2CSS6 | SCI31SCI | SCI31SCO | | | — | — | — | — |
| P5_8 | SENT1RX | TAUD012 | TAUD013 | | TAUD013 | TAUD013 | TAUD013 | TAUD013 | TAUD012 | CSIH2CSS7 | SCI32RX/ INTP2 | | | | — | — | — | — |
| P5_9 | TAUD014 | TAUD014 | TAUD015 | | TAUD015 | TAUD015 | TAUD015 | TAUD015 | PSI51DIN | | CSIH2SI | SCI32TX | | | ✓ | ✓ | ✓ | ✓ |
| P5_10 | RLIN30RX/ INTP3 | TAUD012 | SENT1RX | SENT1SPCO | SENT1RX | SENT1RX | ADCGTRG0 | ADCG0CNV0 | TAUD012 | PSI51DOUT | SCI32SCI | SCI32SCO | | | ✓ | ✓ | ✓ | ✓ |
| P5_11 | | SENT3SPCO | SENT3RX | TAUD1010 | | | | | CSIH2SSI | | | | | | — | — | — | — |
| P5_12 | RLIN31RX/ INTP4 | SENT2SPCO | SENT2RX | TAUD1011 | | | | | CSIH2RY1 | | | | | | — | — | — | — |
| P5_13 | | SENT2SPCO | SENT2TX | TAUD1012 | TAUD013 | TAUD013 | TAUD013 | TAUD013 | TAUD012 | CSIH2RYO | | | | | — | — | — | — |
| P5_14 | TAUJ000 | SENT3SPCO | SENT3RX | TAUD1013 | ADCGTRG1 | ADCG0CNV1 | ADCG0CNV1 | ADCG0CNV1 | PSI50DIN | CSIH2SO | RLIN30RX/ INTP3 | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| P5_15 | RLIN31RX/ INTP4 | SENT3SPCO | | | | | | | CSIH2SCI | CSIH2SCO | | RLIN30TX | | | — | — | — | — |

2.4.1.9 Port JP0 (JP0)

Table 2.49 Port JP0 (JP0)

| Port Mode (JPMC0_m = 0) | Alternative Mode (JPMC0_m = 1) | | | | | | | | | | | | | | Device | | | |
|-------------------------|--------------------------------|--------|-----------------|--------|--------------------|------------------|-----------------|--------|-----------------|--------|-----------------|--------|--------------------|--------|---------|-----|---------|-----|
| | 1st Alternative | | 2nd Alternative | | 3rd Alternative | | 4th Alternative | | 5th Alternative | | 6th Alternative | | Dedicated Function | | 100-pin | | 144-pin | |
| | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | Input | Output | eVR | DPS | eVR | DPS |
| JP0_0 | | | | | FLSCI3RXD (FPDR) | FLSCI3TXD (FPDT) | | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| JP0_1 | | | | | | FLSCI3TXD (FPDT) | | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| JP0_2 | | | | | FLSCI3SCKI (FPCKI) | | | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| JP0_3 | | | | | | | | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| JP0_4 | | | | | | | | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| JP0_5 | | | | | | | | | | | | | | | ✓ | ✓ | ✓ | ✓ |

2.5 DNF

Digital Noise Filter (DNF) eliminates digital noise from external input signals. This product includes two sorts of DNF: peripheral function DNF and edge detection DNF.

2.5.1 Example of Noise Elimination

Figure 2.9 shows an example of noise elimination in peripheral function DNF and edge detection DNF. In this example, the sampling clock, the sampling count, and the current output level are set to 1/2 of the DNF input clock, two (twice), and low, respectively. “○” in the figure means that high level is detected.

In input examples 1, 2, and 3, in which the same level is detected twice consecutively, the change in the input signal is output internally.

In input examples 4, 5, and 6, in which the same level is not detected twice consecutively, input signal is regarded as noise and eliminated. Therefore, no change in the internal output is found.

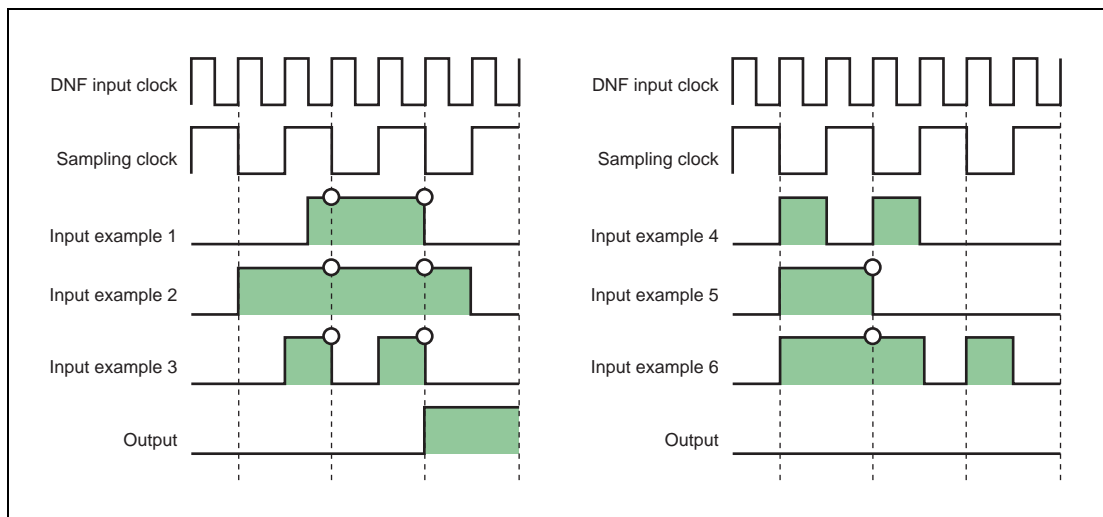


Figure 2.9 Timing Chart of Digital Noise Elimination

2.6 Noise Filter and Edge Level Detection Circuit

Some signals that are input to a pin pass the filter to eliminate noises and glitches. This product supports both analog filter and digital filter.

Additionally, the product also supports the function of detecting an edge or level after a pass through a filter.

The first section describes port input signals to which filters are allocated and the types of the filters, the noise filter and edge level detection control register and its control bits, and an overview of the register address and others.

For details about the digital/analog filter function and the noise filter and edge level detection control register, see **Section 2.6.1, Allocation of Port Filters**.

2.6.1 Allocation of Port Filters

The following is a list of input pins that have analog filter or digital filter.

Table 2.50 List of Input Pins with a Noise Filter (1/6)

| Input Pin | Filter Type | Digital Noise Elimination Control Register | | Digital Noise Elimination Sampling Clock Source Selection Register | | Digital Noise Elimination Enable Register | | | Filter Control Register | | Device | |
|------------------|-------------|--|------------|--|---------|---|-------------|-----------|-------------------------|---------|---------|---------|
| | | Control Register | Address | Control Register | Address | Control Register | Control Bit | Address | Control Register | Address | eVR DPS | eVR DPS |
| INTP0/SCI30RX | Type A | DNFA0CTL | FFC3 0000H | DNFCKS100C.CKSCn[1:0] | DNFA0EN | DNFA0NFENL0 | FFC3 0004H | FCLA1CTL0 | FFC3 4020H | ✓ | ✓ | ✓ |
| INTP1/SCI31RX | Type A | DNFA0CTL | FFC3 0000H | 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB | DNFA0EN | DNFA0NFENL1 | FFC3 0004H | FCLA1CTL1 | FFC3 4024H | ✓ | ✓ | ✓ |
| INTP2/SCI32RX | Type A | DNFA0CTL | FFC3 0000H | 10: High-speed peripheral clock CLK_HSB/2 | DNFA0EN | DNFA0NFENL2 | FFC3 0004H | FCLA1CTL2 | FFC3 4028H | ✓ | ✓ | ✓ |
| INTP3/RLIN30RX | Type A | DNFA1CTL | FFC3 0100H | 11: Setting prohibited | DNFA1EN | DNFA1NFENL0 | FFC3 0104H | FCLA1CTL3 | FFC3 402C | ✓ | ✓ | ✓ |
| INTP4/RLIN31RX | Type A | DNFA1CTL | FFC3 0100H | | DNFA1EN | DNFA1NFENL1 | FFC3 0104H | FCLA1CTL4 | FFC3 4030H | ✓ | ✓ | ✓ |
| INTP5/RSCAN0RX0 | Type A | DNFA2CTL | FFC3 0200H | | DNFA2EN | DNFA2NFENL0 | FFC3 0204H | FCLA1CTL5 | FFC3 4034H | ✓ | ✓ | ✓ |
| INTP6/RSCAN0RX1 | Type A | DNFA2CTL | FFC3 0200H | | DNFA2EN | DNFA2NFENL1 | FFC3 0204H | FCLA1CTL6 | FFC3 4038H | ✓ | ✓ | ✓ |
| INTP7/TAPA0ESO | Type B | DNFA3CTL | FFC3 0300H | | DNFA3EN | DNFA3NFENL0 | FFC3 0304H | FCLA1CTL7 | FFC3 403C | ✓ | ✓ | ✓ |
| INTP8/TAPA1ESO | Type B | DNFA3CTL | FFC3 0300H | | DNFA3EN | DNFA3NFENL1 | FFC3 0304H | FCLA2CTL0 | FFC3 4040H | ✓ | ✓ | ✓ |
| INTP9 | Type C | DNFA4CTL | FFC3 0400H | | DNFA4EN | DNFA4NFENL0 | FFC3 0404H | FCLA2CTL1 | FFC3 4044H | ✓ | ✓ | ✓ |
| INTP10/RSCAN0RX2 | Type A | DNFA2CTL | FFC3 0200H | | DNFA2EN | DNFA2NFENL2 | FFC3 0204H | FCLA2CTL2 | FFC3 4048H | ✓ | ✓ | ✓ |
| INTP11/FLXA0RXDA | Type A | DNFA2CTL | FFC3 0200H | | DNFA2EN | DNFA2NFENL3 | FFC3 0204H | FCLA2CTL3 | FFC3 404C | ✓ | ✓ | ✓ |
| INTP12/FLXA0RXDB | Type A | DNFA2CTL | FFC3 0200H | | DNFA2EN | DNFA2NFENL4 | FFC3 0204H | FCLA2CTL4 | FFC3 4050H | ✓ | ✓ | ✓ |
| NMI | Type C | DNFA4CTL | FFC3 0400H | | DNFA4EN | DNFA4NFENL1 | FFC3 0404H | FCLA0CTL0 | FFC3 4000H | ✓ | ✓ | ✓ |

Table 2.50 List of Input Pins with a Noise Filter (2/6)

| Input Pin | Filter Type | Digital Noise Elimination Control Register | | Digital Noise Elimination Sampling Clock Source Selection Register | | Digital Noise Elimination Enable Register | | | Filter Control Register | | Device | | | |
|-----------|-------------|--|------------|---|------------|---|-------------|------------|-------------------------|---------|--------|-----|-----|-----|
| | | Control Register | Address | Control Register | Address | Control Register | Control Bit | Address | Control Register | Address | eVR | DPS | eVR | DPS |
| TAUD010 | Type E | DNFA5CTL | FFC3 0500H | DNFCKSnC.CKSCn[1:0] | FFC3 0504H | DNFA5EN | DNFA5NFENL0 | FFC3 0504H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD011 | Type E | DNFA5CTL | FFC3 0500H | 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: TAUD0TCKEN0 | FFC3 0504H | DNFA5EN | DNFA5NFENL1 | FFC3 0504H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD012 | Type E | DNFA6CTL | FFC3 0600H | | FFC3 0604H | DNFA6EN | DNFA6NFENL0 | FFC3 0604H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD013 | Type E | DNFA6CTL | FFC3 0600H | | FFC3 0604H | DNFA6EN | DNFA6NFENL1 | FFC3 0604H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD014 | Type E | DNFA7CTL | FFC3 0700H | | FFC3 0704H | DNFA7EN | DNFA7NFENL0 | FFC3 0704H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD015 | Type E | DNFA7CTL | FFC3 0700H | | FFC3 0704H | DNFA7EN | DNFA7NFENL1 | FFC3 0704H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD016 | Type E | DNFA8CTL | FFC3 0800H | | FFC3 0804H | DNFA8EN | DNFA8NFENL0 | FFC3 0804H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD017 | Type E | DNFA8CTL | FFC3 0800H | | FFC3 0804H | DNFA8EN | DNFA8NFENL1 | FFC3 0804H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD018 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL0 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD019 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL1 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD010 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL2 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD011 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL3 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD012 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL4 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD013 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL5 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD014 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL6 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |
| TAUD015 | Type E | DNFA9CTL | FFC3 0900H | | FFC3 0904H | DNFA9EN | DNFA9NFENL7 | FFC3 0904H | — | — | ✓ | ✓ | ✓ | ✓ |

Table 2.50 List of Input Pins with a Noise Filter (3/6)

| Input Pin | Filter Type | Digital Noise Elimination Control Register | | Digital Noise Elimination Enable Register | | Digital Noise Elimination Sampling Clock Source Selection Register | | Filter Control Register | | Device | |
|-----------|-------------|--|------------|---|------------|--|------------|-------------------------|---------|---------|---------|
| | | Control Register | Address | Control Register | Address | Control Register | Address | Control Register | Address | 100-pin | 144-pin |
| TAUD110 | Type E | DNFA10CTL | FFC3 0A00H | DNFA10EN | FFC3 0A04H | DNFA10EN | FFC3 0A04H | — | — | ✓ | ✓ |
| TAUD111 | Type E | DNFA10CTL | FFC3 0A00H | DNFA10EN | FFC3 0A04H | DNFA10EN | FFC3 0A04H | — | — | ✓ | ✓ |
| TAUD112 | Type E | DNFA11CTL | FFC3 0B00H | DNFA11EN | FFC3 0B04H | DNFA11EN | FFC3 0B04H | — | — | ✓ | ✓ |
| TAUD113 | Type E | DNFA11CTL | FFC3 0B00H | DNFA11EN | FFC3 0B04H | DNFA11EN | FFC3 0B04H | — | — | ✓ | ✓ |
| TAUD114 | Type E | DNFA12CTL | FFC3 0C00H | DNFA12EN | FFC3 0C04H | DNFA12EN | FFC3 0C04H | — | — | — | ✓ |
| TAUD115 | Type E | DNFA12CTL | FFC3 0C00H | DNFA12EN | FFC3 0C04H | DNFA12EN | FFC3 0C04H | — | — | — | ✓ |
| TAUD116 | Type E | DNFA13CTL | FFC3 0D00H | DNFA13EN | FFC3 0D04H | DNFA13EN | FFC3 0D04H | — | — | ✓ | ✓ |
| TAUD117 | Type E | DNFA13CTL | FFC3 0D00H | DNFA13EN | FFC3 0D04H | DNFA13EN | FFC3 0D04H | — | — | — | ✓ |
| TAUD118 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | ✓ | ✓ |
| TAUD119 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | — | ✓ |
| TAUD110 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | ✓ | ✓ |
| TAUD111 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | — | ✓ |
| TAUD112 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | — | ✓ |
| TAUD113 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | — | ✓ |
| TAUD114 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | — | ✓ |
| TAUD115 | Type E | DNFA14CTL | FFC3 0E00H | DNFA14EN | FFC3 0E04H | DNFA14EN | FFC3 0E04H | — | — | — | ✓ |

Table 2.50 List of Input Pins with a Noise Filter (4/6)

| Input Pin | Filter Type | Digital Noise Elimination Control Register | | Digital Noise Elimination Enable Register | Filter Control Register | | Device | | | |
|-----------|-------------|--|------------|---|-------------------------|------------|---------|---------|---|---|
| | | Control Register | Address | | Control Register | Address | 100-pin | 144-pin | | |
| TAUD210 | Type E | DNFA15CTL | FFC3 0F00H | DNFA15EN | DNFA15NFENL0 | FFC3 0F04H | — | ✓ | ✓ | ✓ |
| TAUD211 | Type E | DNFA15CTL | FFC3 0F00H | DNFA15EN | DNFA15NFENL1 | FFC3 0F04H | — | ✓ | ✓ | ✓ |
| TAUD212 | Type E | DNFA16CTL | FFC3 1000H | DNFA16EN | DNFA16NFENL0 | FFC3 1004H | — | ✓ | ✓ | ✓ |
| TAUD213 | Type E | DNFA16CTL | FFC3 1000H | DNFA16EN | DNFA16NFENL1 | FFC3 1004H | — | ✓ | ✓ | ✓ |
| TAUD214 | Type E | DNFA17CTL | FFC3 1100H | DNFA17EN | DNFA17NFENL0 | FFC3 1104H | — | — | — | ✓ |
| TAUD215 | Type E | DNFA17CTL | FFC3 1100H | DNFA17EN | DNFA17NFENL1 | FFC3 1104H | — | — | — | ✓ |
| TAUD216 | Type E | DNFA18CTL | FFC3 1200H | DNFA18EN | DNFA18NFENL0 | FFC3 1204H | — | — | — | ✓ |
| TAUD217 | Type E | DNFA18CTL | FFC3 1200H | DNFA18EN | DNFA18NFENL1 | FFC3 1204H | — | ✓ | ✓ | ✓ |
| TAUD218 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL0 | FFC3 1304H | — | — | — | ✓ |
| TAUD219 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL1 | FFC3 1304H | — | — | — | ✓ |
| TAUD210 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL2 | FFC3 1304H | — | — | — | ✓ |
| TAUD211 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL3 | FFC3 1304H | — | ✓ | ✓ | ✓ |
| TAUD212 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL4 | FFC3 1304H | — | ✓ | ✓ | ✓ |
| TAUD213 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL5 | FFC3 1304H | — | ✓ | ✓ | ✓ |
| TAUD214 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL6 | FFC3 1304H | — | ✓ | ✓ | ✓ |
| TAUD215 | Type E | DNFA19CTL | FFC3 1300H | DNFA19EN | DNFA19NFENL7 | FFC3 1304H | — | ✓ | ✓ | ✓ |
| TAUJ010 | Type E | DNFA20CTL | FFC3 1400H | DNFA20EN | DNFA20NFENL0 | FFC3 1404H | — | ✓ | ✓ | ✓ |
| TAUJ011 | Type E | DNFA21CTL | FFC3 1500H | DNFA21EN | DNFA21NFENL0 | FFC3 1504H | — | ✓ | ✓ | ✓ |
| TAUJ012 | Type E | DNFA22CTL | FFC3 1600H | DNFA22EN | DNFA22NFENL0 | FFC3 1604H | — | ✓ | ✓ | ✓ |
| TAUJ013 | Type E | DNFA23CTL | FFC3 1700H | DNFA23EN | DNFA23NFENL0 | FFC3 1704H | — | ✓ | ✓ | ✓ |
| TAUJ110 | Type E | DNFA24CTL | FFC3 1800H | DNFA24EN | DNFA24NFENL0 | FFC3 1804H | — | ✓ | ✓ | ✓ |
| TAUJ111 | Type E | DNFA25CTL | FFC3 1900H | DNFA25EN | DNFA25NFENL0 | FFC3 1904H | — | ✓ | ✓ | ✓ |
| TAUJ112 | Type E | DNFA26CTL | FFC3 1A00H | DNFA26EN | DNFA26NFENL0 | FFC3 1A04H | — | ✓ | ✓ | ✓ |
| TAUJ113 | Type E | DNFA27CTL | FFC3 1B00H | DNFA27EN | DNFA27NFENL0 | FFC3 1B04H | — | ✓ | ✓ | ✓ |
| TAUJ210 | Type E | DNFA28CTL | FFC3 1C00H | DNFA28EN | DNFA28NFENL0 | FFC3 1C04H | — | ✓ | ✓ | ✓ |
| TAUJ211 | Type E | DNFA29CTL | FFC3 1D00H | DNFA29EN | DNFA29NFENL0 | FFC3 1D04H | — | ✓ | ✓ | ✓ |
| TAUJ212 | Type E | DNFA30CTL | FFC3 1E00H | DNFA30EN | DNFA30NFENL0 | FFC3 1E04H | — | ✓ | ✓ | ✓ |
| TAUJ213 | Type E | DNFA31CTL | FFC3 1F00H | DNFA31EN | DNFA31NFENL0 | FFC3 1F04H | — | ✓ | ✓ | ✓ |

Table 2.50 List of Input Pins with a Noise Filter (5/6)

| Input Pin | Filter Type | Digital Noise Elimination Control Register | | Digital Noise Elimination Sampling Clock Source Selection Register | Digital Noise Elimination Enable Register | | | Filter Control Register | | Device | |
|--------------------|-------------|--|------------|---|---|--------------|------------|-------------------------|------------|------------|---------|
| | | Control Register | Address | | Control Register | Control Bit | Address | Control Register | Address | 100-pin | 144-pin |
| TSG30PTS10/ENCA0E0 | Type E | DNFA32CTL | FFC3 2000H | DNFCKS104C.CKSCn[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA32EN | DNFA32NFENL0 | FFC3 2004H | — | — | ✓ | ✓ |
| TSG30PTS11/ENCA0E1 | Type E | DNFA32CTL | FFC3 2000H | DNFCKS105C.CKSC105[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA32EN | DNFA32NFENL1 | FFC3 2004H | — | — | ✓ | ✓ |
| TSG30PTS12/ENCA0E2 | Type E | DNFA33CTL | FFC3 2100H | | DNFA33EN | DNFA33NFENL0 | FFC3 2104H | — | — | — | ✓ |
| TSG30CLKI | Type E | DNFA34CTL | FFC3 2200H | DNFCKS106C.CKSC106[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA34EN | DNFA34NFENL0 | FFC3 2204H | — | — | ✓ | ✓ |
| TSG31PTS10/ENCA1E0 | Type E | DNFA35CTL | FFC3 2300H | | DNFA35EN | DNFA35NFENL0 | FFC3 2304H | — | — | ✓ | ✓ |
| TSG31PTS11/ENCA1E1 | Type E | DNFA35CTL | FFC3 2300H | DNFCKS107C.CKSC107[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA35EN | DNFA35NFENL1 | FFC3 2304H | — | — | ✓ | ✓ |
| TSG31PTS12/ENCA1E2 | Type E | DNFA36CTL | FFC3 2400H | | DNFA36EN | DNFA36NFENL0 | FFC3 2404H | — | — | ✓ | ✓ |
| TSG31CLKI | Type E | DNFA37CTL | FFC3 2500H | DNFCKS108C.CKSC108[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA37EN | DNFA37NFENL0 | FFC3 2504H | — | — | ✓ | ✓ |
| ENCA010 | Type E | DNFA38CTL | FFC3 2600H | | DNFA38EN | DNFA38NFENL0 | FFC3 2604H | — | — | ✓ | ✓ |
| ENCA011 | Type E | DNFA38CTL | FFC3 2600H | DNFCKS109C.CKSC109[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA38EN | DNFA38NFENL1 | FFC3 2604H | — | — | ✓ | ✓ |
| ENCA110 | Type E | DNFA39CTL | FFC3 2700H | | DNFA39EN | DNFA39NFENL0 | FFC3 2704H | — | — | ✓ | ✓ |
| ENCA111 | Type E | DNFA39CTL | FFC3 2700H | DNFCKS110C.CKSC110[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA39EN | DNFA39NFENL1 | FFC3 2704H | — | — | ✓ | ✓ |
| CSIG0RYI | Type F | DNFA40CTL | FFC3 2800H | | DNFA40EN | DNFA40NFENL0 | FFC3 2804H | — | — | — | ✓ |
| CSIH0RYI | Type F | DNFA40CTL | FFC3 2800H | DNFCKS111C.CKSC111[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA40EN | DNFA40NFENL1 | FFC3 2804H | — | — | ✓ | ✓ |
| CSIH1RYI | Type F | DNFA40CTL | FFC3 2800H | | DNFA40EN | DNFA40NFENL2 | FFC3 2804H | — | — | ✓ | ✓ |
| CSIH2RYI | Type F | DNFA40CTL | FFC3 2800H | DNFCKS112C.CKSC112[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA40EN | DNFA40NFENL3 | FFC3 2804H | — | — | ✓ | ✓ |
| CSIH3RYI | Type F | DNFA40CTL | FFC3 2800H | | DNFA40EN | DNFA40NFENL4 | FFC3 2804H | — | — | ✓ | ✓ |
| ADCGTRG0 | Type F | DNFA41CTL | FFC3 2900H | DNFCKS113C.CKSC113[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA41EN | DNFA41NFENL0 | FFC3 2904H | — | — | ✓ | ✓ |
| ADCGTRG1 | Type F | DNFA41CTL | FFC3 2900H | | DNFA41EN | DNFA41NFENL1 | FFC3 2904H | — | — | ✓ | ✓ |
| SENT0RX | Type D | DNFA42CTL | FFC3 2A00H | DNFCKS114C.CKSC114[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA42EN | DNFA42NFENL0 | FFC3 2A04H | FCLA3CTL0 | FFC3 4060H | ✓ | ✓ |
| SENT1RX | Type D | DNFA42CTL | FFC3 2A00H | | DNFA42EN | DNFA42NFENL1 | FFC3 2A04H | FFC3 4064H | FCLA3CTL1 | FFC3 4064H | ✓ |
| SENT2RX | Type D | DNFA42CTL | FFC3 2A00H | DNFCKS115C.CKSC115[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA42EN | DNFA42NFENL2 | FFC3 2A04H | FCLA3CTL2 | FFC3 4068H | — | ✓ |
| SENT3RX | Type D | DNFA42CTL | FFC3 2A00H | | DNFA42EN | DNFA42NFENL3 | FFC3 2A04H | FFC3 4072H | FCLA3CTL3 | FFC3 4068H | ✓ |
| SENT4RX | Type D | DNFA42CTL | FFC3 2A00H | DNFCKS116C.CKSC116[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA42EN | DNFA42NFENL4 | FFC3 2A04H | FCLA3CTL4 | FFC3 4076H | ✓ | ✓ |
| SENT5RX | Type D | DNFA42CTL | FFC3 2A00H | | DNFA42EN | DNFA42NFENL5 | FFC3 2A04H | FFC3 4080H | FCLA3CTL5 | FFC3 4076H | ✓ |

Table 2.50 List of Input Pins with a Noise Filter (6/6)

| Input Pin | Filter Type | Digital Noise Elimination Control Register | | Digital Noise Elimination Sampling Clock Source Selection Register (DNFCKSnC.CKSCn[1:0]) | Digital Noise Elimination Enable Register | | | Filter Control Register | | Device | | | |
|---------------------------|-------------|--|------------|---|---|--------------|------------|-------------------------|------------|---------|---------|-----|-----|
| | | Control Register | Address | | Control Register | Control Bit | Address | Control Register | Address | 100-pin | 144-pin | | |
| PSI50DIN | Type D | DNFA43CTL | FFC3 2B00H | DNFCKS107C.CKSC107[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA43EN | DNFA43NFENL0 | FFC3 2B04H | FCLA4CTL0 | FFC3 4080H | eVR | DPS | eVR | DPS |
| PSI51DIN | Type D | DNFA43CTL | FFC3 2B00H | DNFCKS107C.CKSC107[1:0] 00: Setting prohibited 01: High-speed peripheral clock CLK_HSB 10: High-speed peripheral clock CLK_HSB/2 11: Setting prohibited | DNFA43EN | DNFA43NFENL1 | FFC3 2B04H | FCLA4CTL1 | FFC3 4084H | eVR | DPS | eVR | DPS |
| $\overline{\text{RESET}}$ | ANF | — | — | — | — | — | — | — | — | — | — | — | — |
| DCUTRST | ANF | — | — | — | — | — | — | — | — | — | — | — | — |
| FLMDO | ANF | — | — | — | — | — | — | — | — | — | — | — | — |

2.6.2 Filter Type

2.6.2.1 Digital Filter Type A (INTP/RXD Alternative) Input Pin

The input pin of digital filter type A has a digital filter and the edge detection function. The digital filter and edge detection are controlled with the following registers. Interrupt signals and data received signals of peripheral functions are shared.

- Filter control register FCLAnCTLm
Ports with each digital filter have the dedicated FCLAnCTLm register.
- Digital noise elimination control register DNFACTL
Each DNFACTL control register controls digital filter processing.
- Digital noise elimination enable register DNFAEN
The register enables or disables digital noise elimination.

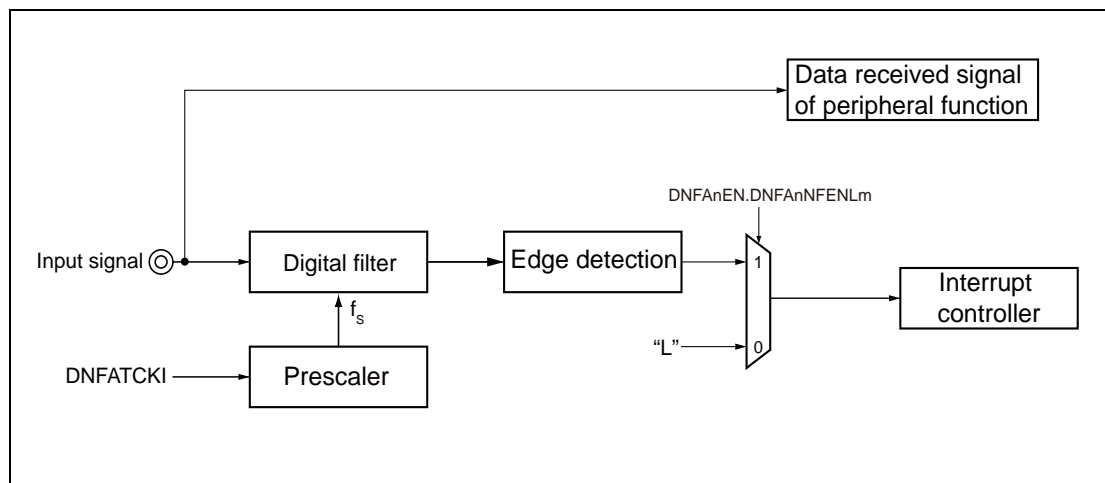


Figure 2.10 Block Diagram of Digital Filter Type A

CAUTION

Set DNFAEN.DNFAENFENLm to 1 to enable input to interrupt controller.

2.6.2.2 Digital Filter Type B (INTP/ESO Alternative) Input Pin

The input pin of digital filter type B has a digital filter, analog filter, and the edge detection function. The digital filter and edge detection are controlled with the following registers. Interrupt signals and Hi-Z output control signals are shared.

- Filter control register FCLAnCTLm
Ports with each digital filter have the dedicated FCLAnCTLm register.
- Digital noise elimination control register DNFACTL
Each DNFACTL control register controls digital filter processing.
- Digital noise elimination enable register DNFAEN
The register selects a digital filter or analog filter.

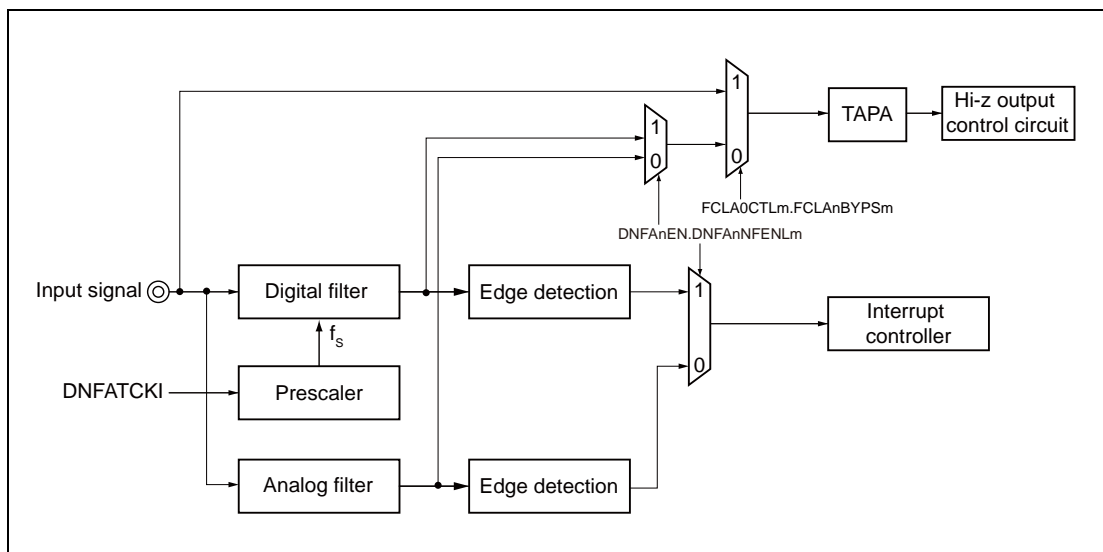


Figure 2.11 Block Diagram of Digital Filter Type B

2.6.2.3 Digital Filter Type C (for INTP9 and NMI) Input Pin

The input pin of digital filter type C has a digital filter and the edge detection function. The digital filter and edge detection are controlled with the following registers.

- Filter control register FCLAnCTLm
Ports with each digital filter have the dedicated FCLAnCTLm register.
- Digital noise elimination control register DNFAAnCTL
Each DNFAAnCTL control register controls digital filter processing.
- Digital noise elimination enable register DNFAAnEN
The register enables or disables digital noise elimination.

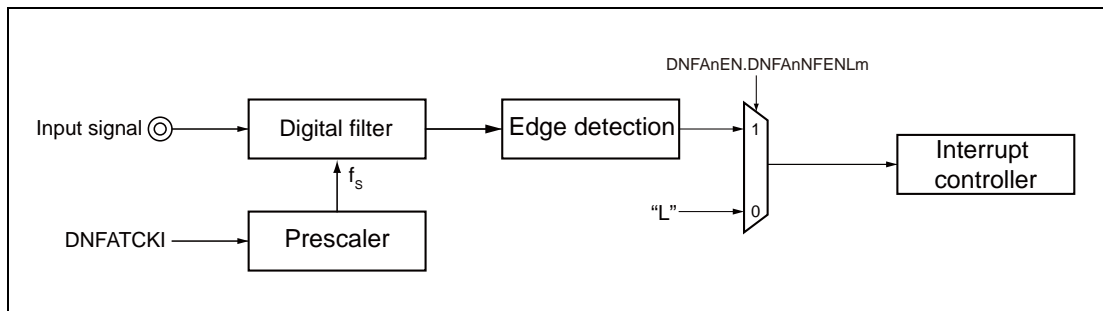


Figure 2.12 Block Diagram of Digital Filter Type C

CAUTION

Set DNFAAnEN.DNFAAnNFENLm to 1 to enable input to interrupt controller.

2.6.2.4 Digital Filter Type D (for SENT, PSI5) Input Pin

The input pin of digital filter type D has a digital filter and analog filter. The digital filter and analog filter are controlled with the following registers.

- Filter control register FCLAnCTLm
The dedicated FCLAnCTLm register is available for each port with digital filter.
- Digital noise elimination control register DNFAAnCTL
Each DNFAAnCTL control register controls digital filter processing.
- Digital noise elimination enable register DNFAAnEN
The register selects a digital filter or analog filter.

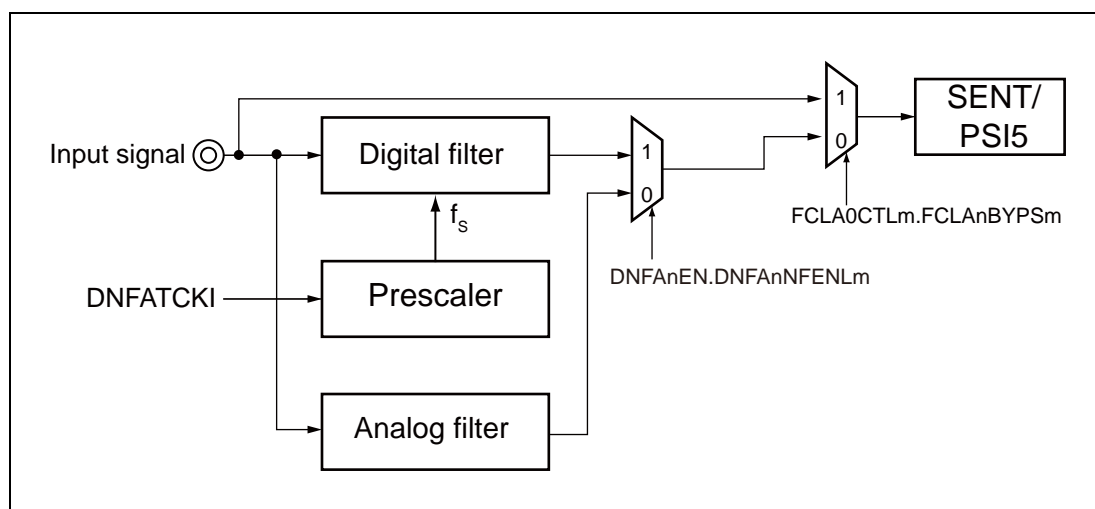


Figure 2.13 Block Diagram of Digital Filter Type D

2.6.2.5 Digital Filter Type E (No Edge Detection) Input Pin

The input pin of digital filter type E has a digital filter. The digital filter is controlled with the following register.

- Digital noise elimination control register DNFA_nCTL
Each DNFA_nCTL control register controls digital filter processing.
- Digital noise elimination enable register DNFA_nEN
This register enables or disables digital noise elimination.

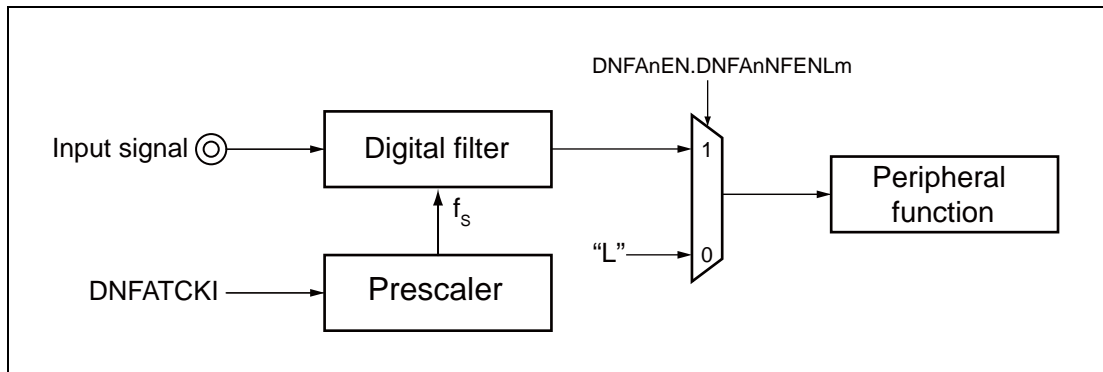


Figure 2.14 Block Diagram of Digital Filter Type E

2.6.2.6 Digital Filter Type F (No Edge Detection) Input Pin

The input pin of digital filter type F has a digital filter. The digital filter is controlled with the following register.

- Digital noise elimination control register DNFA_nCTL
Each DNFA_nCTL control register controls digital filter processing.
- Digital noise elimination enable register DNFA_nEN
This register enables or disables digital noise elimination.

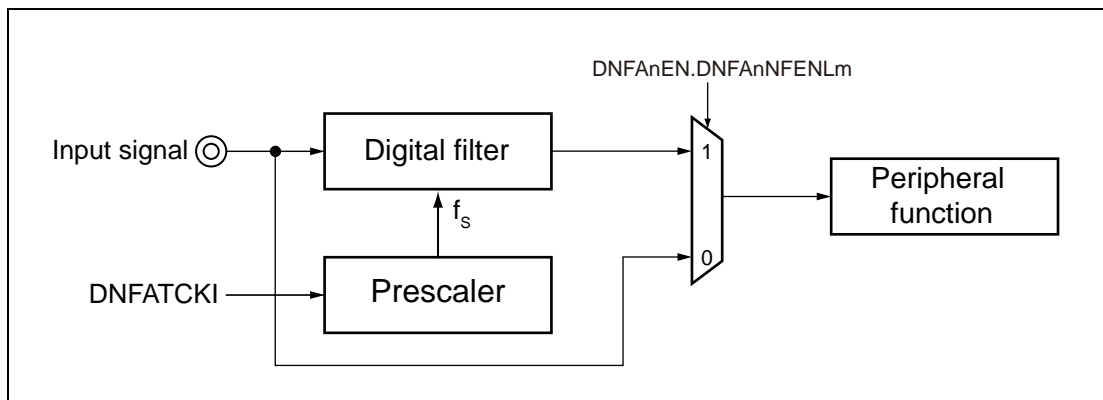


Figure 2.15 Block Diagram of Digital Filter Type F

2.6.2.7 Analog Filter ANF Input Pin

The analog filter input pin has only an analog filter.

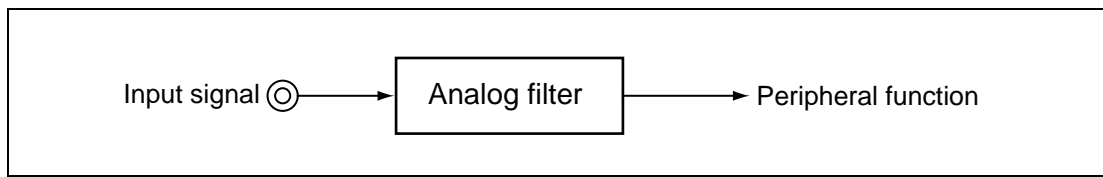


Figure 2.16 Block Diagram of Analog Filter ANF

2.6.3 Registers

2.6.3.1 List of Registers

The digital noise filter is controlled and operated by the following registers.

Table 2.51 List of Digital Noise Filter Registers

| Register Name | Symbol | Address |
|---|-------------|--|
| Filter control register | FCLAnCTLm | See Table 2.50, List of Input Pins with a Noise Filter. |
| Digital noise elimination control register | DNFAnCTL | FFC3 0000 _H + n × 100 _H |
| Digital noise elimination enable register | DNFAnEN | FFC3 0000 _H + n × 100 _H + 4 _H |
| Digital noise elimination enable register L | DNFAnENL | FFC3 0000 _H + n × 100 _H + C _H |
| Digital noise elimination sampling clock source select register | DNFCKSnC | See Section 2.6.3.6, DNFCKSnC — Digital Noise Elimination Sampling Clock Source Selection Register. |
| Digital noise elimination sampling clock source status register | DNFCSCnSTAT | See Section 2.6.3.7, DNFCSCnSTAT — Digital Noise Elimination Sampling Clock Source Status Register. |

CAUTIONS

1. Do not change any setting of any related control register (FCLAnCTLm, DNFAnCTL, DNFAnEN, DNFAnENL, and DNFCKSnC) while the digital filter is enabled (DNFAnEN.DNFAnNFENLm = 1). Attempting to change any of the settings may lead to unexpected output from the filter.
2. Each of the registers has bits to which no function is assigned. Unless otherwise specified, do not write any value other than the initial value to these bits. Correct operation cannot be guaranteed if writing of other values is attempted.

2.6.3.2 FCLAnCTLm — Filter Control Register

This register controls the analog and digital filter operation.

Operations differ according to the filter type. For details, see **Section 2.6.2, Filter Type**.

Access: This register can be read/written in 8-bit or 1-bit units.

Address: See **Table 2.50, List of Input Pins with a Noise Filter**.

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|---|---|---|---|---|------------|------------|
| | FCLAnBYPSm | — | — | — | — | — | FCLAnINTFm | FCLAnINTRm |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R | R | R/W | R/W |

Table 2.52 FCLAnCTLm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 7 | FCLAnBYPSm | Filter and bypass control of the alternative function 0: Filters and bypasses are disabled. 1: Filters and bypasses are enabled. |
| 6 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | FCLAnINTFm | Falling edge detection control 0: Falling edge detection disabled 1: Falling edge detection enabled |
| 0 | FCLAnINTRm | Rising edge detection control 0: Rising edge detection disabled 1: Rising edge detection enabled |

NOTE

For details about the relevant Pins, see **Table 2.50, List of Input Pins with a Noise Filter**.

2.6.3.3 DNFACTL — Digital Noise Elimination Control Register

This register specifies characteristics of the digital noise elimination filter.

Access: This register can be read/written in 8-bit units.

Address: See Table 2.50, List of Input Pins with a Noise Filter.

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|-----------------|-----|---|---|---------------|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | DNFAnNFSTS[1:0] | | — | — | DNFAnPRS[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R | R | R/W | R/W | R/W |

Table 2.53 DNFACTL Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | |
|------------------|--------------------|---|-----------------|--------------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|---------------|------------------|---------------|------------------|---------------|------------------|----------------|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | |
| 6, 5 | DNFAnNFSTS [1:0] | The DNFAnNFSTS[1:0] bits specify the number of samples used to judge whether an external signal pulse is valid. <table border="1" data-bbox="678 907 1417 1097"> <thead> <tr> <th>DNFAnNFSTS[1:0]</th> <th>Number of Samples</th> </tr> </thead> <tbody> <tr> <td>00_B</td> <td>2</td> </tr> <tr> <td>01_B</td> <td>3</td> </tr> <tr> <td>10_B</td> <td>4</td> </tr> <tr> <td>11_B</td> <td>5</td> </tr> </tbody> </table> | DNFAnNFSTS[1:0] | Number of Samples | 00 _B | 2 | 01 _B | 3 | 10 _B | 4 | 11 _B | 5 | | | | | | | | |
| DNFAnNFSTS[1:0] | Number of Samples | | | | | | | | | | | | | | | | | | | |
| 00 _B | 2 | | | | | | | | | | | | | | | | | | | |
| 01 _B | 3 | | | | | | | | | | | | | | | | | | | |
| 10 _B | 4 | | | | | | | | | | | | | | | | | | | |
| 11 _B | 5 | | | | | | | | | | | | | | | | | | | |
| 4, 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | |
| 2 to 0 | DNFAnPRS [2:0] | Selects a digital filter sampling clock frequency <table border="1" data-bbox="678 1232 1417 1563"> <thead> <tr> <th>DNFAnPRS[2:0]</th> <th>Sampling Frequency</th> </tr> </thead> <tbody> <tr> <td>000_B</td> <td>DNFATCKI / 1</td> </tr> <tr> <td>001_B</td> <td>DNFATCKI / 2</td> </tr> <tr> <td>010_B</td> <td>DNFATCKI / 4</td> </tr> <tr> <td>011_B</td> <td>DNFATCKI / 8</td> </tr> <tr> <td>100_B</td> <td>DNFATCKI / 16</td> </tr> <tr> <td>101_B</td> <td>DNFATCKI / 32</td> </tr> <tr> <td>110_B</td> <td>DNFATCKI / 64</td> </tr> <tr> <td>111_B</td> <td>DNFATCKI / 128</td> </tr> </tbody> </table> | DNFAnPRS[2:0] | Sampling Frequency | 000 _B | DNFATCKI / 1 | 001 _B | DNFATCKI / 2 | 010 _B | DNFATCKI / 4 | 011 _B | DNFATCKI / 8 | 100 _B | DNFATCKI / 16 | 101 _B | DNFATCKI / 32 | 110 _B | DNFATCKI / 64 | 111 _B | DNFATCKI / 128 |
| DNFAnPRS[2:0] | Sampling Frequency | | | | | | | | | | | | | | | | | | | |
| 000 _B | DNFATCKI / 1 | | | | | | | | | | | | | | | | | | | |
| 001 _B | DNFATCKI / 2 | | | | | | | | | | | | | | | | | | | |
| 010 _B | DNFATCKI / 4 | | | | | | | | | | | | | | | | | | | |
| 011 _B | DNFATCKI / 8 | | | | | | | | | | | | | | | | | | | |
| 100 _B | DNFATCKI / 16 | | | | | | | | | | | | | | | | | | | |
| 101 _B | DNFATCKI / 32 | | | | | | | | | | | | | | | | | | | |
| 110 _B | DNFATCKI / 64 | | | | | | | | | | | | | | | | | | | |
| 111 _B | DNFATCKI / 128 | | | | | | | | | | | | | | | | | | | |

NOTE

For details about the relevant Pins, see Table 2.50, List of Input Pins with a Noise Filter.

2.6.3.4 DNFA_nEN — Digital Noise Elimination Enable Register

This register enables or disables digital noise elimination for an arbitrary input signal.

Operations differ according to the filter type. For details, see **Section 2.6.2, Filter Type**.

Access: This register can be read/written in 16-bit units.

Address: See **Table 2.50, List of Input Pins with a Noise Filter**.

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| | — | — | — | — | — | — | — | — | DNFA _n NFENL7 | DNFA _n NFENL6 | DNFA _n NFENL5 | DNFA _n NFENL4 | DNFA _n NFENL3 | DNFA _n NFENL2 | DNFA _n NFENL1 | DNFA _n NFENL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 2.54 DNFA_nEN Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------------------|---|
| 15 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 to 0 | DNFA _n NFENL [7:0] | Digital Noise Elimination Control 0: Digital noise elimination disabled (Analog noise elimination is enabled.)*1 1: Digital noise elimination enabled |

Note 1. For digital filter Type B and D

NOTE

- For details about the relevant Pins, see **Table 2.50, List of Input Pins with a Noise Filter**.
- Bits DNFA_nNFENL[7:0] which are not listed in **Table 2.59, List of Input Pins with a Noise Filter** are reserved. When read, the value after reset is read. When writing, write the value after reset.

2.6.3.5 DNFA_nENL — Digital Noise Elimination Enable Register

This register corresponds to the lower 8 bits of the DNFA_nEN register.

Access: This register can be read/written 8- and 1- bit units.

Address: FFC3 0000_H + n × 100_H + C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| | DNFA _n NFENL7 | DNFA _n NFENL6 | DNFA _n NFENL5 | DNFA _n NFENL4 | DNFA _n NFENL3 | DNFA _n NFENL2 | DNFA _n NFENL1 | DNFA _n NFENL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

For details of the respective bit functions, see **Section 2.6.3.4, DNFA_nEN — Digital Noise Elimination Enable Register**.

2.6.3.6 DNFCKSnC — Digital Noise Elimination Sampling Clock Source Selection Register

This register selects the clock source for digital noise sampling clock frequency (DNFATCKI). (n = 100 to 114)

Access: This register can be read / written in 32-bit units.

Address: DNFCKS100C FFF8 9120_H
 DNFCKS101C FFF8 9128_H
 DNFCKS106C FFF8 9130_H
 DNFCKS107C FFF8 9138_H
 DNFCKS108C FFF8 9140_H
 DNFCKS109C FFF8 9148_H
 DNFCKS110C FFF8 9150_H
 DNFCKS112C FFF8 9158_H
 DNFCKS113C FFF8 9160_H
 DNFCKS114C FFF8 9168_H
 DNFCKS104C FFF8 9170_H
 DNFCKS105C FFF8 9178_H
 DNFCKS102C FFF8 9180_H

Value after reset: 0000 0002_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CKSCn [1:0] | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R |

Table 2.55 DNFCKSnC Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|------------|--|--------|--------|-----------------------|---|---|--------------------|---|---|---|---|---|--|---|---|--|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 2, 1 | CKSCn[1:0] | These bits specify the sampling frequency of the source clock used to judge whether an external signal pulse is valid. <table border="1" data-bbox="678 1438 1417 1626"> <thead> <tr> <th>CKSCn1</th> <th>CKSCn0</th> <th>Sampling Clock Source</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>Selects a clock source for digital filter sampling clock frequency (DNFATCKI). DNFATCKI differs depending on the relevant input signal. For details, see Table 2.50, List of Input Pins with a Noise Filter.</td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table> | CKSCn1 | CKSCn0 | Sampling Clock Source | 0 | 0 | Setting prohibited | 0 | 1 | Selects a clock source for digital filter sampling clock frequency (DNFATCKI). DNFATCKI differs depending on the relevant input signal. For details, see Table 2.50, List of Input Pins with a Noise Filter. | 1 | 0 | | 1 | 1 | |
| CKSCn1 | CKSCn0 | Sampling Clock Source | | | | | | | | | | | | | | | |
| 0 | 0 | Setting prohibited | | | | | | | | | | | | | | | |
| 0 | 1 | Selects a clock source for digital filter sampling clock frequency (DNFATCKI). DNFATCKI differs depending on the relevant input signal. For details, see Table 2.50, List of Input Pins with a Noise Filter. | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | | | | | | |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |

2.6.3.7 DNFCSCnSTAT — Digital Noise Elimination Sampling Clock Source Status Register

This register is used to check the status of the sampling clock source for digital noise sampling clock (DNFATCKI). (n = 100 to 114)

Access: This register can be read only in 32-bit units.

Address: DNFCSC100STAT FFF8 9124_H
 DNFCSC101STAT FFF8 912C_H
 DNFCSC106STAT FFF8 9134_H
 DNFCSC107STAT FFF8 913C_H
 DNFCSC108STAT FFF8 9144_H
 DNFCSC109STAT FFF8 914C_H
 DNFCSC110STAT FFF8 9154_H
 DNFCSC112STAT FFF8 915C_H
 DNFCSC113STAT FFF8 9164_H
 DNFCSC114STAT FFF8 916C_H
 DNFCSC104STAT FFF8 9174_H
 DNFCSC105STAT FFF8 917C_H
 DNFCSC102STAT FFF8 9184_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|--------------|---------|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CLKSELn[1:0] | CLKACTn | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 2.56 DNFCSCnSTAT Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | |
|--------------|--------------|---|--------|--------|-----------------------|---|---|--------------------|---|---|---|---|---|---|---|
| 31 to 3 | Reserved | When read, the value after reset is read. | | | | | | | | | | | | | |
| 2, 1 | CLKSELn[1:0] | Sampling clock source selected can be checked. <table border="1" data-bbox="678 1400 1417 1590"> <thead> <tr> <th>CKSCn1</th> <th>CKSCn0</th> <th>Sampling clock source</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td rowspan="3">Clock source (DNFATCKI) selected by the DNFCSCnC Register can be checked. DNFATCKI differs depending on the relevant input signal. For details, see Table 2.50, List of Input Pins with a Noise Filter.</td> </tr> <tr> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table> | CKSCn1 | CKSCn0 | Sampling clock source | 0 | 0 | Setting prohibited | 0 | 1 | Clock source (DNFATCKI) selected by the DNFCSCnC Register can be checked. DNFATCKI differs depending on the relevant input signal. For details, see Table 2.50, List of Input Pins with a Noise Filter . | 1 | 0 | 1 | 1 |
| CKSCn1 | CKSCn0 | Sampling clock source | | | | | | | | | | | | | |
| 0 | 0 | Setting prohibited | | | | | | | | | | | | | |
| 0 | 1 | Clock source (DNFATCKI) selected by the DNFCSCnC Register can be checked. DNFATCKI differs depending on the relevant input signal. For details, see Table 2.50, List of Input Pins with a Noise Filter . | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | | | | |
| 0 | CLKACTn | Operation of clock source selected by the DNFCSCnC Register can be checked. 0: The clock source stops 1: The clock source in operation | | | | | | | | | | | | | |

2.6.4 Notes for Digital Noise Filter

- (1) When digital filter output signal is input for alternative function:

Enable the digital filter (DNFAnEN.DNFAnENm (m = 0 to 7) = 1), and switch the port pin to alternative function after a lapse of the following time.

$$s = \text{DNFAnNFSTS}[1:0] + 2$$

$$\text{Time} = s \times 1/\text{fs} + 2 \times 1/\text{DNFATCKI}$$

- (2) When an event output signal of the digital filter is used as an interrupt:

Enable the digital filter (DNFAnEN.DNFAnENm (m = 0 to 7) = 1 while interrupt is disabled.

Enable the digital filter and clear the interrupt request flag after a lapse of the following time, then enable the interrupt.

$$\text{Time} = s \times 1/\text{fs} + 3 \times 1/\text{DNFATCKI}$$

2.7 Pin State

Table 2.57, Table 2.58, and Table 2.59 show detailed pin states. Some pins may not be included depending on grades and packages of the product. For the pins included in each product, see Table 2.1, Pin Assignment, in Section 2.1, Pin Connection Diagrams.

Table 2.57 Pin State in Single Chip Mode (Normal Mode)

| Port | Pin Category | OPJTAG/ OPEVTI/ OPEVTO | Power OFF | DCUTRST = L | | | | |
|----------|--------------------------|----------------------------------|--------------|--------------------------------------|--------------------|------------|--|-----------|
| | | | | RESET pin = L Power ON | Reset Release*1 | Field BIST | RUN (User program) | CVM Reset |
| GPIO | | — | — | Hi-Z | Hi-Z | Hi-Z | Operate | Hi-Z |
| JP0_x | JP0_0 to JP0_3, JP0_5 | OPJTAG = 00 | — | Hi-Z | Hi-Z | Hi-Z | Operate | Hi-Z |
| | DCUTDI | OPJTAG = 11 | — | | | | | |
| | DCUTDO | | | | | | | |
| | DCUTCK | | | | | | | |
| | DCUTMS | | | | | | | |
| | DCUTRDY | OPJTAG = 01 | — | | | | | |
| | LPDI | | | | | | | |
| | LPDO | | | | | | | |
| | LPDCLK | | | | | | | |
| | JP0_3 LPDCLKOUT | | | | | | | |
| P0_13 | GPIO (EVTI) | OPEVTI = 1 | — | Hi-Z | | | Operate | Hi-Z |
| | EVTI | OPEVTI = 0 | — | | | | | |
| RESET | | — | — | L (PD) | H (PD) | H (PD) | H (PD) | H (PD) |
| ERROROUT | | — | — | Hi-Z before RESET is released. | L | L | Operate | Hi-Z |
| P0_10 | RESETOUT (/GPIO) | OPEVTO = 1 or DCUTRST = L | — | L | L | L | H*4 (set "High" by user program) | L |
| | EVT0 | OPEVTO = 0 and DCUTRST = H | — | N.A. | N.A. | N.A. | N.A. | N.A. |
| CVMOUT | | — | — | L → H*2 | H | H | H | L |
| FLMD0*3 | | — | — | L (PD) | L (PD) | L (PD) | L (PD) | L (PD) |
| P3_14 | FLMD1(/GPIO) | — | — | Hi-Z | Hi-Z | Hi-Z | Operate*4 | Hi-Z |
| X1/X2 | | — | — | Stop → Run | Run | Run | Run | Run |
| ANI | | — | — | Hi-Z | Hi-Z | Hi-Z | Operate | Hi-Z |
| DCUTRST | | — | — | L (PD) | L (PD) | L (PD) | L (PD) | L (PD) |

Note 1. The states listed in the table are those following release from a pin reset, ECM reset, SW reset, or CVM reset.

Note 2. Once CVMOUT is driven high, it continues to reflect the voltage status even if the RESET pin is driven low.

Note 3. Drive FLMD0 low.

Note 4. This pin functions as a port pin.

Table 2.58 Pin State in Single Chip Mode (Debug Mode)

| Port | Pin Category | OPJTAG/ OPEVTI/ OPEVTO | DCUTRST = H | | | | |
|---------------------|-----------------------|---|---|---------------|-----------------------------|--------------------------|-----------|
| | | | RUN | RESET pin = L | Reset Release* ¹ | Debugger Initiated Reset | CVM Reset |
| GPIO | | — | Operate | Hi-Z | Hi-Z | Hi-Z | Hi-Z |
| JP0_x | JP0_0 to JP0_3, JP0_5 | OPJTAG = 00 | N.A. | | | N.A. | N.A. |
| | DCUTDI | OPJTAG = 11 | Operate (If JTAG is selected based on OPJTAG setting.) | | | Operate | Operate |
| | DCUTDO | | | | | | |
| | DCUTCK | | | | | | |
| | DCUTMS | | | | | | |
| | DCUTRDY | | | | | | |
| | LPDI | OPJTAG = 01 | Operate (If LPD4P is selected based on OPJTAG setting.) | | | Operate | Operate |
| | LPDO | | | | | | |
| | LPDCLK | | | | | | |
| | JP0_3 | | Operate | Hi-Z | Hi-Z | | |
| LPDCLKOUT | | Operate (If LPD4P is selected based on OPJTAG setting.) | | | Operate | Operate | |
| P0_13 | GPIO (EVTI) | OPEVTI = 1 | Operate | Hi-Z | Hi-Z | Hi-Z | Hi-Z |
| | EVTI | OPEVTI = 0 | Operate (If EVTI is selected based on OPEVTI setting.) | | | Operate | Operate |
| RESET | — | H (PD) | L (PD) | H (PD) | H (PD) | H (PD) | |
| ERROROUT | — | Operate | Hi-Z | L | L | Hi-Z | |
| P0_10 | RESETOUT (/GPIO) | OPEVTO = 1 or DCUTRST = L | H* ² (Set to "High" by user program) | L | L | L | L |
| | EVTO | OPEVTO = 0 and DCUTRST = H | Operate (If EVTO is selected based on OPJTAG setting.) | | | Operate | Operate |
| CVMOUT | — | H | H | H | L | L | |
| FLMD0* ³ | — | L (PD) | L (PD) | L (PD) | L (PD) | L (PD) | |
| P3_14 | FLMD1(/GPIO) | — | Operate* ² | Hi-Z | Hi-Z | Hi-Z | Hi-Z |
| X1/X2 | — | Run | Run | Run | Run | Run | |
| ANI | — | Operate | Hi-Z | Hi-Z | Hi-Z | Hi-Z | |
| DCUTRST | — | H (PD) | H (PD) | H (PD) | H (PD) | H (PD) | |

Note 1. The states listed in the table are those following release from a pin reset, debug reset, ECM reset, SW reset, or CVM reset.

Note 2. This pin functions as a port pin.

Note 3. Drive FLMD0 low.

Table 2.59 Pin State in Serial Programming Mode

| Port | Pin Category | OPJTAG/ OPEVTI/ OPEVTO | Power OFF | DCUTRST = L | | |
|----------|--|------------------------------|--------------|---------------|------------------|---|
| | | | | RESET pin = L | | RUN (Serial Programming Mode) |
| | | | | Power ON | Reset Release | |
| GPIO | | — | — | Hi-Z | Hi-Z | Operate |
| JP0_x | JTAG PORT (FLSCI3RXD (FPDR)/ FLSCI3TXD (FPDT)/ FLSCI3SCKI (FPCK)) | OPJTAG = xx | — | Hi-Z | Hi-Z | Operate |
| P0_13 | GPIO (EVTI) | OPEVTI = x | — | Hi-Z | Hi-Z | Operate as GPIO (not depend on OPEVTI setting) |
| RESET | | — | — | L (PD) | H (PD) | H (PD) |
| ERROROUT | | — | — | Hi-Z | L | Operate |
| P0_10 | RESETOUT/(GPIO) | OPEVTO = x or DCUTRST = x | — | L | L | L |
| CVMOUT | | — | — | L → H | H (masked) *1 | H (masked) *1 |
| FLMD0*2 | | — | — | H (PD) | H (PD) | H (PD) |
| P3_14 | FLMD1/(GPIO)*2 | — | — | L | L | L |
| X1/X2 | | — | — | Stop → Run | Run | Run |
| ANI | | — | — | Hi-Z | Hi-Z | Operate |
| DCUTRST | | — | — | L (PD) | L (PD) | L (PD) |

Note 1. CVMOUT is masked and outputs high level in serial programming mode.

Note 2. Drive the FLMD0 pin to the high level and the FLMD1 pin to the low level.

2.8 Handling of Unused Pins

The alternative functions can be selected in the option bytes, please refer **Section 35.10.3, OPBT2 — Option Byte 2**.

We recommend use of pull-up/-down resistor with the value of 1 kΩ or above.

When pull-up/-down resistor is used, connect a resistor between two pins. Do not directly connect the pins with a wire.

Table 2.60 Handling of Unused Pins (JP0)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|---|
| JP0_0 | GPI/O | Input: When input buffer is disabled in input port mode (JPIBC0_m=0 and JPMC0_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (JPIBC0_m = 1 or JPMC0_m = 1): Connected to E1VCC or E1VSS through resistor Output: Open |
| | DCUTDI | Connected to E1VCC through resistor |
| | LPDI | Open |
| JP0_1 | GPI/O | Input: When input buffer is disabled in input port mode (JPIBC0_m=0 and JPMC0_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (JPIBC0_m=1 or JPMC0_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| | DCUTDO | Open |
| | LPDO | Open |
| JP0_2 | GPI | Input: When input buffer is disabled in input port mode (JPIBC0_m=0 and JPMC0_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (JPIBC0_m=1 or JPMC0_m=1): Connected to E1VCC or E1VSS through resistor |
| | DCUTCK | Open |
| | LPDCLK | Open |
| JP0_3 | GPI/O | Input: When input buffer is disabled in input port mode (JPIBC0_m=0 and JPMC0_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (JPIBC0_m=1 or JPMC0_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| | DCUTMS | Connected to E1VCC through resistor |
| JP0_4 | DCUTRST | Connected to VSS |
| JP0_5 | GPI/O | Input: When input buffer is disabled in input port mode (JPIBC0_m=0 and JPMC0_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (JPIBC0_m=1 or JPMC0_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| | DCUTRDY | Open |
| | LPDCLKOUT | Open |

Table 2.61 Handling of Unused Pins (P0) (1/2)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|---|
| P0_0 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open |
| P0_1 | | |
| P0_2 | | When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_3 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_4 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_5 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_6 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_7 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_8 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_9 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |

Table 2.61 Handling of Unused Pins (P0) (2/2)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|--------------------------|---|
| P0_10 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VSS through resistor Output: Open |
| | RESETOUT | |
| | $\overline{\text{EVTO}}$ | Open |
| P0_11 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P0_12 | | |
| P0_13 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| | | |
| P0_14 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |

Table 2.62 Handling of Unused Pins (P1)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|---|
| P1_0 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E0VCC or E0VSS through resistor Output: Open |
| P1_1 | | |
| P1_2 | | |
| P1_3 | | |
| P1_4 | | |

Table 2.63 Handling of Unused Pins (P2)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|---|
| P2_0 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E0VCC or E0VSS through resistor Output: Open |
| P2_1 | | |
| P2_2 | | |
| P2_3 | | |
| P2_4 | | |
| P2_5 | | |
| P2_6 | | |
| P2_7 | | |
| P2_8 | | |
| P2_9 | | |
| P2_10 | | |
| P2_11 | | |
| P2_12 | | |
| P2_13 | | |
| P2_14 | | |
| P2_15 | | |

Table 2.64 Handling of Unused Pins (P3)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|---|
| P3_0 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E0VCC or E0VSS through resistor Output: Open |
| P3_1 | | |
| P3_2 | | |
| P3_3 | | |
| P3_4 | | |
| P3_5 | | |
| P3_6 | | |
| P3_7 | | |
| P3_8 | | |
| P3_9 | | |
| P3_10 | | |
| P3_11 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E1VCC or E1VSS through resistor Output: Open |
| P3_12 | | |
| P3_13 | | |
| P3_14 | GPI/O | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E0VCC or E0VSS through resistor Output: Open |
| P3_15 | | |
| FLMD1 | | Connected to E0VSS through resistor |

Table 2.65 Handling of Unused Pins (P4)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|---|
| P4_0 | GPIO | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E0VCC or E0VSS through resistor Output: Open |
| P4_1 | | |
| P4_2 | | |
| P4_3 | | |
| P4_4 | | |
| P4_5 | | |
| P4_6 | | |
| P4_7 | | |
| P4_8 | | |
| P4_9 | | |
| P4_10 | | |
| P4_11 | | |
| P4_12 | | |
| P4_13 | | |
| P4_14 | | |

Table 2.66 Handling of Unused Pins (P5)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|---|
| P5_0 | GPIO | Input: When input buffer is disabled in input port mode (PIBCn_m=0 and PMCN_m=0): Open When input buffer is enabled in input port mode or input is selected in S/W I/O control alternative mode (PIBCn_m=1 or PMCN_m=1): Connected to E0VCC or E0VSS through resistor Output: Open |
| P5_1 | | |
| P5_2 | | |
| P5_3 | | |
| P5_4 | | |
| P5_5 | | |
| P5_6 | | |
| P5_7 | | |
| P5_8 | | |
| P5_9 | | |
| P5_10 | | |
| P5_11 | | |
| P5_12 | | |
| P5_13 | | |
| P5_14 | | |
| P5_15 | | |

Table 2.67 Handling of Unused Pins (Others)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|-------------------------------------|
| FLMD0 | — | Connected to E1VSS through resistor |
| RESET | — | This pin must be used. |
| CVMOUT | — | Open |
| ERROROUT | — | Open |
| X1 | — | This pin must be used. |
| X2 | — | This pin must be used. |
| VCC | — | This pin must be used. |
| VSS | — | This pin must be used. |
| VDD | — | This pin must be used. |
| VCL | — | This pin must be used. |
| EnVCC | — | This pin must be used. |
| EnVSS | — | This pin must be used. |
| A0VCC | — | This pin must be used. |
| A0VREFH | — | This pin must be used. |
| A0VSS | — | This pin must be used. |
| A1VCC | — | This pin must be used. |
| A1VREFH | — | This pin must be used. |
| A1VSS | — | This pin must be used. |

Table 2.68 Handling of Unused Pins (ADCG0, ADCG1)

| Pin Name | Alternative Function | Handling of Unused Pins |
|----------|----------------------|------------------------------|
| ADCG010 | ADIN | Connected to A0VCC or A0VSS. |
| ADCG011 | | |
| ADCG012 | | |
| ADCG013 | | |
| ADCG014 | | |
| ADCG015 | | |
| ADCG016 | | |
| ADCG017 | | |
| ADCG018 | | |
| ADCG019 | | |
| ADCG0110 | | |
| ADCG0111 | | |
| ADCG110 | ADIN | Connected to A1VCC or A1VSS. |
| ADCG111 | | |
| ADCG112 | | |
| ADCG113 | | |
| ADCG114 | | |
| ADCG115 | | |
| ADCG116 | | |
| ADCG117 | | |
| ADCG118 | | |
| ADCG119 | | |
| ADCG1110 | | |
| ADCG1111 | | |

Section 3 CPU System

3.1 Overview

3.1.1 Block Configuration

Figure 3.1 shows the block configuration diagram.

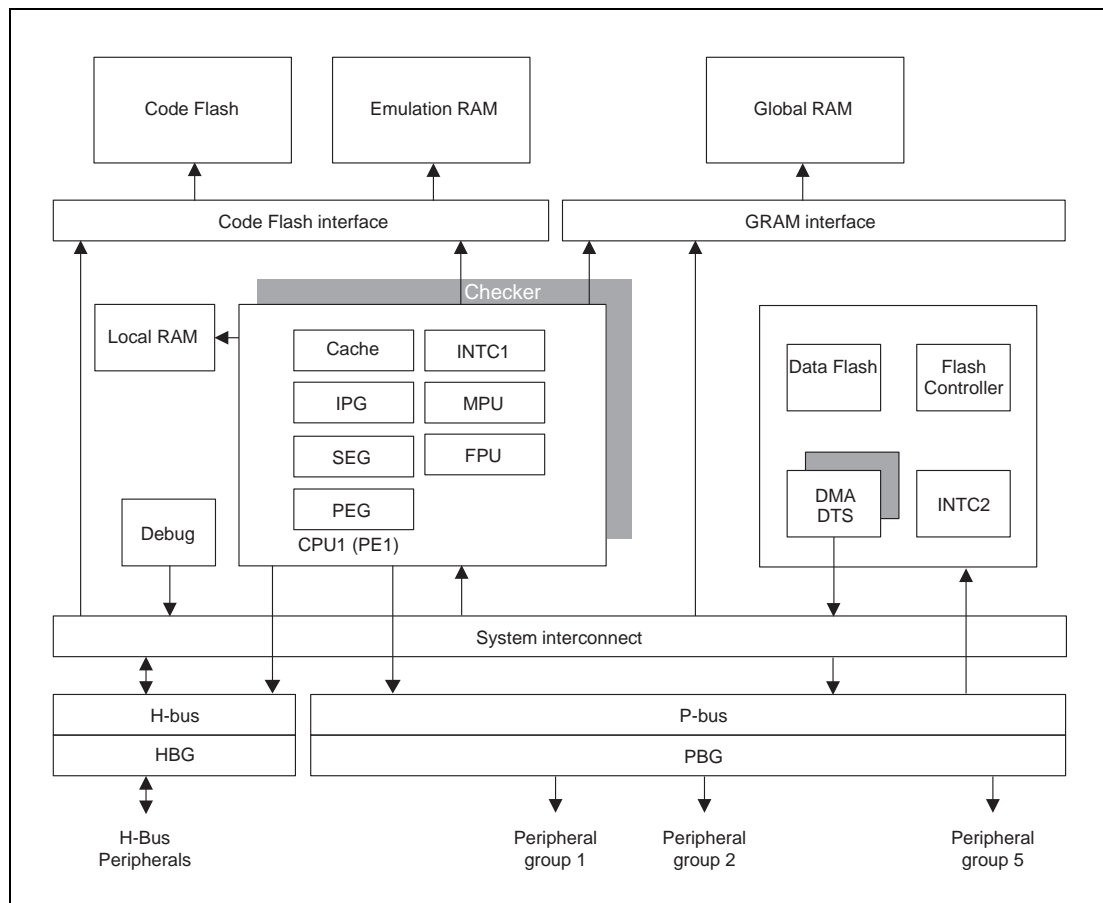


Figure 3.1 Block Configuration Diagram

CPU1 (PE1)

The RH850 G3M core is included as a main CPU. CPU1 also includes the checker core for safety assurance.

Local RAM

A high-speed accessible local RAM is included.

Global RAM

The Global RAM is a mass on-chip RAM that all PEs can share.

Code Flash

A mass Code Flash is included for program storage. CPU1 and the Code Flash are connected with each other via the Code Flash interface.

Emulation RAM

It is a RAM to emulate the Code Flash. The programs can be replaced without rewriting the Code Flash through control from an external tool.

Data Flash

It is a Flash memory being rewritable by the CPU.

P-Bus and H-Bus

The P-bus and H-bus connect the peripheral IPs. The P-bus is divided into peripheral groups 1, 2 and 5.

INTC1, INTC2

The interrupt controller consists of INTC1 and INTC2. INTC1 has a redundant configuration with the checker core.

DMA

Two DMA transfer modules, DMAC and DTS, are included. The DMA is provided with a checker for safety.

Slave Guard

The slave guard is a system to prevent unauthorized access from the specific bus master, consisting of the following guard structures.

(1) PE Guard (PEG)

The PE guard is a system to prevent unauthorized access to the resources in the PE from the external master. After reset release, access is prohibited except access from the PE itself.

(2) Internal Peripheral Guard (IPG)

Each PE has an “Internal Peripheral Guard” (IPG) that protects the registers of peripherals inside the PE against unauthorized accesses.

(3) System Error Generation (SEG)

The registers controls how to response SYSERR.

(4) Global RAM Guard (GRG)

The global RAM guard is a system to prevent unauthorized access to the global RAM from the external master. After reset release, the global RAM is unprotected (accessible from all bus masters).

(5) Peripheral Guard (PBG, HBG)

The peripheral guard is a system to prevent unauthorized access to peripheral devices from the external master. Peripheral devices are in the protected state excluding PE1 after release from the reset state.

3.2 CPU

3.2.1 Core Functions

3.2.1.1 Features

Table 3.1 lists features of the RH850G3M core.

Table 3.1 Features of the RH850G3M Core

| Item | Feature |
|-------------------------|---|
| CPU | <ul style="list-style-type: none"> • Advanced 32-bit architecture for embedded control • 32-bit internal data bus • Thirty-two 32-bit general registers • RISC-type instruction sets <ul style="list-style-type: none"> – Long-/short-format load/store instructions – Three-operand instructions – Instruction sets based on C language • CPU operating modes <ul style="list-style-type: none"> – User mode and supervisor mode • Address space: 4-Gbyte linear address space for both data and instructions • Instructions: A snooze instruction (SNOOZE) is included for temporary suspension by switching the CPU clock signal (CLK_CPU) off for 32 clock cycles. |
| Coprocessor | <ul style="list-style-type: none"> • A floating-point operation coprocessor (FPU) mounted <ul style="list-style-type: none"> – Supports single precision (32 bits) and double precision (64 bits). – Supports data types and exceptions conforming to IEEE754. – Rounding mode: Neighborhood, 0 direction, +∞ direction, and -∞ direction – Handling denormalized numbers: Rounding down to 0 or exception notification to conform to IEEE754 |
| Exception/ Interrupt | <ul style="list-style-type: none"> • 16 interrupt priority levels settable for each channel • Vector selection method selectable according to performance request or memory usage <ul style="list-style-type: none"> – Direct branching exception vectors – Indirect branching exception vectors referring to the address table – Supports the high-speed save/restore processing of the context by the dedicated instructions (PUSHSP and POPSP) at the generation of an interrupt |
| Memory management | <ul style="list-style-type: none"> • Memory protection function (MPU): 16 areas settable |
| Cache | <ul style="list-style-type: none"> • Instruction cache |

3.2.1.2 Register Set

This subsection explains the program registers and system registers incorporated in this CPU.

(1) Program Registers

Program registers include the general-purpose registers (r0 to r31) and program counter (PC).

r0 always retains 0, whereas the value after reset is undefined in r1 to r31.

Table 3.2 List of Program Registers

| Program Register | Name | Function | Description |
|---------------------------|-----------|--|--|
| General-purpose registers | r0 | Zero register | Always retains "0" |
| | r1 | Assembler reserved register | Used as working register for generating addresses |
| | r2 | Register for address and data variables (used when the real-time OS used does not use this register) | |
| | r3 | Stack pointer (SP) | Used for generating a stack frame when a function is called |
| | r4 | Global pointer (GP) | Used for accessing a global variable in the data area |
| | r5 | Text pointer (TP) | Used as a register that indicates the start of the text area (area where program code is placed) |
| | r6 to r29 | Register for addresses and data variables | |
| | r30 | Element pointer (EP) | Used as a base pointer for generating addresses when accessing memory |
| | r31 | Link pointer (LP) | Used when the compiler calls a function |
| Program counter | PC | Retains instruction addresses during execution of programs | |

NOTE

For further descriptions of r1, r3 to r5, and r31 used for an assembler and/or C compiler, see the specification of each software development environment.

(a) General-Purpose Registers

A total of 32 general-purpose registers (r0 to r31) are provided. All of these registers can be used for either data variables or address variables.

Of the general-purpose registers, r0 to r5, r30, and r31 are assumed to be used for special purposes in software development environments, so it is necessary to note the following when using them.

1. r0, r3, and r30

These registers are implicitly used by instructions.

r0 is a register that always retains “0”. It is used for operations that use 0 and addressing with base address being 0.

r3 is implicitly used by the PREPARE, DISPOSE, PUSHSP, and POPSP instructions.

r30 is used as a base pointer when the SLD instruction or SST instruction accesses memory.

2. r1, r4, r5, and r31

These registers are implicitly used by the assembler and C compiler.

When using these registers, register contents must first be saved so they are not lost and can be restored after the registers are used.

3. r2

This register might be used by a real-time OS in some cases. If the real-time OS that is being used is not using r2, r2 can be used as a register for address variables or data variables.

(b) PC — Program Counter

The PC retains the address of the instruction being executed. Bit 0 is fixed to 0, and branching to an odd number address is disabled.

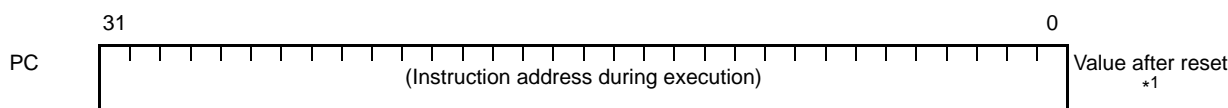


Table 3.3 PC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 1 | PC31-1 | These bits indicate the address of the instruction being executed. | R/W | *1 |
| 0 | PC0 | This bit is fixed to 0, and branching to an odd number address is disabled. | R/W | 0 |

Note 1. The value after reset differs depending on the setting value of the reset vector. For details, see (q) RBASE — Reset vector base address register.

(2) Basic System Registers

The basic system registers are used to control CPU status and to retain exception information.

System registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.4 Basic System Registers

| Register No. (regID, selID) | Name | Function | Access Permission |
|--------------------------------|--------|---|----------------------|
| SR0, 0 | EIPC | Status save registers when acknowledging EI level exception | SV |
| SR1, 0 | EIPSW | Status save registers when acknowledging EI level exception | SV |
| SR2, 0 | FEPC | Status save registers when acknowledging FE level exception | SV |
| SR3, 0 | FEPSW | Status save registers when acknowledging FE level exception | SV |
| SR5, 0 | PSW | Program status word | *1 |
| SR6, 0 | FPSR | Refer to (4) FPU Function Registers. | CU0 and SV |
| SR7, 0 | FPEPC | Refer to (4) FPU Function Registers. | CU0 and SV |
| SR8, 0 | FPST | Refer to (4) FPU Function Registers. | CU0 |
| SR9, 0 | FPCC | Refer to (4) FPU Function Registers. | CU0 |
| SR10, 0 | FPCFG | Refer to (4) FPU Function Registers. | CU0 |
| SR11, 0 | FPEC | Refer to (4) FPU Function Registers. | CU0 and SV |
| SR13, 0 | EIIC | EI level exception cause | SV |
| SR14, 0 | FEIC | FE level exception cause | SV |
| SR16, 0 | CTPC | CALLT execution status save register | UM |
| SR17, 0 | CTPSW | CALLT execution status save register | UM |
| SR20, 0 | CTBP | CALLT base pointer | UM |
| SR28, 0 | EIWR | EI level exception working register | SV |
| SR29, 0 | FEWR | FE level exception working register | SV |
| SR31, 0 | BSEL | Not implemented. A value of 0 is returned when read and writing is ignored. | SV |
| SR0, 1 | MCFG0 | Machine configuration | SV |
| SR2, 1 | RBASE | Reset vector base address | SV |
| SR3, 1 | EBASE | Exception handler vector address | SV |
| SR4, 1 | INTBP | Base address of the interrupt handler table | SV |
| SR5, 1 | MCTL | CPU control | SV |
| SR6, 1 | PID | Processor ID | SV |
| SR11, 1 | SCCFG | SYSCALL operation setting | SV |
| SR12, 1 | SCBP | SYSCALL base pointer | SV |
| SR0, 2 | HTCFG0 | Thread configuration | SV |
| SR6, 2 | MEA | Memory error address | SV |
| SR7, 2 | ASID | Address space ID | SV |
| SR8, 2 | MEI | Memory error information | SV |

Note 1. The access permission differs depending on the bit.

(a) EIPC — Status save register when acknowledging EI level exception

When an EI level exception is acknowledged, the address of the instruction that was being executed when the EI level exception occurred, or of the next instruction, is saved to the EIPC register (see “Types of Exceptions” in the *RH850G3M User's Manual: Software*).

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the EIPC register. An odd-numbered address must not be specified.

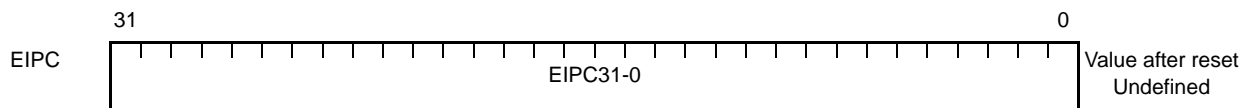


Table 3.5 EIPC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 1 | EIPC31-1 | These bits indicate the PC saved when an EI level exception is acknowledged. | R/W | Undefined |
| 0 | EIPC0 | This bit indicates the PC saved when an EI level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the EIRET instruction is executed is 0. | R/W | Undefined |

(b) EIPSW — Status save register when acknowledging EI level exception

When an EI level exception is acknowledged, the current PSW setting is saved to the EIPSW register.

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

CAUTION

Bits 11 to 9 cannot usually be changed since these bits are associated with debugging.

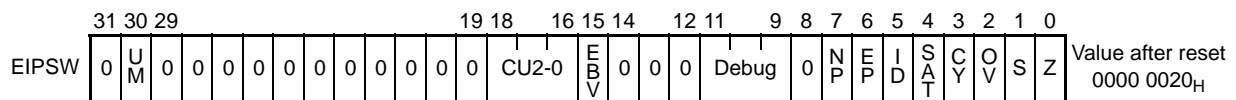


Table 3.6 EIPSW Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 30 | UM | This bit stores the PSW.UM bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 29 to 19 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 18 to 16 | CU2-0 | These bits store the PSW.CU 2 to 0 field setting when an EI level exception is acknowledged. (CU2-1 are reserved for future expansion. Be sure to set to 0.) | R/W | 0 |
| 15 | EBV | This bit stores the PSW.EBV bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 14 to 12 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 11 to 9 | Debug | These bits store the PSW.Debug field setting when an EI level exception is acknowledged. | R/W | 0 |
| 8 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 7 | NP | This bit stores the PSW.NP bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 6 | EP | This bit stores the PSW.EP bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 5 | ID | This bit stores the PSW.ID bit setting when an EI level exception is acknowledged. | R/W | 1 |
| 4 | SAT | This bit stores the PSW.SAT bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 3 | CY | This bit stores the PSW.CY bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 2 | OV | This bit stores the PSW.OV bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 1 | S | This bit stores the PSW.S bit setting when an EI level exception is acknowledged. | R/W | 0 |
| 0 | Z | This bit stores the PSW.Z bit setting when an EI level exception is acknowledged. | R/W | 0 |

(c) FEPC — Status save register when acknowledging FE level exception

When an FE level exception is acknowledged, the address of the instruction that was being executed when the FE level exception occurred, or of the next instruction, is saved to the FEPC register (see “Types of Exceptions” in *RH850G3M User's Manual: Software*).

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the FEPC register. An odd-numbered address must not be specified.

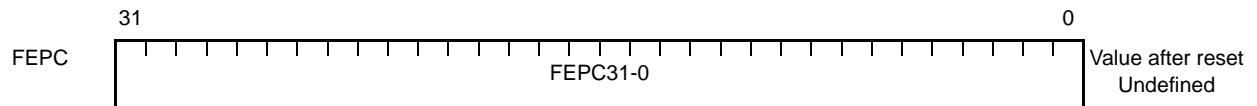


Table 3.7 FEPC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 1 | FEPC31-1 | These bits indicate the PC saved when an FE level exception is acknowledged. | R/W | Undefined |
| 0 | FEPC0 | This bit indicates the PC saved when an FE level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the FERET instruction is executed is 0. | R/W | Undefined |

(d) FEPSW — Status save register when acknowledging FE level exception

When an FE level exception is acknowledged, the current PSW setting is saved to the FEPSW register.

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

CAUTION

Bits 11 to 9 cannot usually be changed since these bits are associated with debugging.

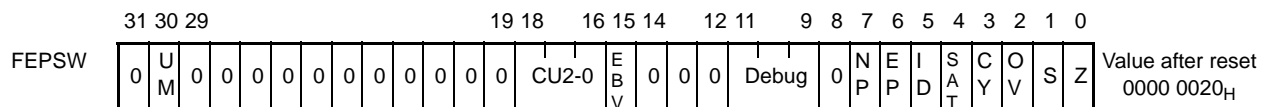


Table 3.8 FEPSW Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 30 | UM | This bit stores the PSW.UM bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 29 to 19 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 18 to 16 | CU2-0 | These bits store the PSW.CU 2 to 0 field setting when an FE level exception is acknowledged. (CU2-1 are reserved for future expansion. Be sure to set to 0.) | R/W | 0 |
| 15 | EBV | This bit stores the PSW.EBV bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 14 to 12 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 11 to 9 | Debug | These bits store the PSW.Debug field setting when an FE level exception is acknowledged. | R/W | 0 |
| 8 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 7 | NP | This bit stores the PSW.NP bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 6 | EP | This bit stores the PSW.EP bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 5 | ID | This bit stores the PSW.ID bit setting when an FE level exception is acknowledged. | R/W | 1 |
| 4 | SAT | This bit stores the PSW.SAT bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 3 | CY | This bit stores the PSW.CY bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 2 | OV | This bit stores the PSW.OV bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 1 | S | This bit stores the PSW.S bit setting when an FE level exception is acknowledged. | R/W | 0 |
| 0 | Z | This bit stores the PSW.Z bit setting when an FE level exception is acknowledged. | R/W | 0 |

(e) PSW — Program status word

PSW (program status word) is a set of flags that indicate the program status (instruction execution result) and bits that indicate the operation status of the CPU (flags are bits in the PSW that are referenced by a condition instruction (Bcond, CMOV, etc.)).

CAUTIONS

1. When the LDSR instruction is used to change the contents of bits 7 to 0 in this register, the changed contents become valid immediately after completion of the LDSR instruction execution.
2. The access permission for the PSW register differs depending on the bit. All bits can be read, but some bits can only be written under certain conditions. See Table 3.9, Access Permission for PSW Register for the access permission for each bit.

Table 3.9 Access Permission for PSW Register

| Bit | Access Permission When Reading | Access Permission When Writing |
|----------|--------------------------------|--------------------------------|
| 30 | UM | SV* ¹ |
| 18 to 16 | CU2-0 | SV* ¹ |
| 15 | EBV | SV* ¹ |
| 11 to 9 | Debug | Special* ¹ |
| 7 | NP | SV* ¹ |
| 6 | EP | SV* ¹ |
| 5 | ID | SV* ¹ |
| 4 | SAT | UM |
| 3 | CY | UM |
| 2 | OV | UM |
| 1 | S | UM |
| 0 | Z | UM |

Note 1. The access permission for the whole PSW register is UM, so the PIE exception does not occur even if the register is written by using an LDSR instruction when PSW.UM is 1. In this case, writing is ignored.

| PSW | 31 | 30 | 29 | 19 | 18 | 16 | 15 | 14 | 12 | 11 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Value after reset | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|---|---|---|-------|-----|---|---|---|-------|---|-------------------|----|----|-----|----|----|---|---|------------------------|
| | 0 | UM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CU2-0 | EBV | 0 | 0 | 0 | Debug | 0 | NP | EP | ID | SAT | CY | OV | S | Z | 0000 0020 _H |

Table 3.10 PSW Register Contents (1/2)

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 30 | UM | This bit indicates that the CPU is in user mode (UM mode). 0: Supervisor mode 1: User mode | R/W | 0 |
| 29 to 19 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 18 to 16 | CU2-CU0 | This bit indicates the coprocessor use permissions. When the bit corresponding to the coprocessor is 0, a coprocessor use prohibition exception is generated in response to execution of a coprocessor instruction or access to coprocessor resources system registers. CU2 bit 18 : (Reserved for future expansion. Be sure to clear to 0.) CU1 bit 17 : (Reserved for future expansion. Be sure to clear to 0.) CU0 bit 16 : FPU | R/W | 0 |

Table 3.10 PSW Register Contents (2/2)

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|-------------------|---|-----|-------------------|
| 15 | EBV | This bit indicates the reset vector and exception vector operation. See the description on RBASE ((q) RBASE — Reset Vector Base Address Register) and EBASE ((r) EBASE — Exception Handler Vector Address Register) in this section. | R/W | 0 |
| 14 to 12 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 11 to 9 | Debug | These bits are used in the debugging functions of development tools. In normal operation, set these bits to 0. | R/W | 0 |
| 8 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 7 | NP | This bit disables the acknowledgement of FE level exception. When an FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level and FE level exceptions. As for the exceptions which the NP bit disables the acknowledgment, see the <i>Exception Cause List of RH850G3M User's Manual: Software</i> . 0: The acknowledgement of FE level exception is enabled. 1: The acknowledgement of FE level exception is disabled. | R/W | 0 |
| 6 | EP | This bit indicates that an exception other than an interrupt controlled by the interrupt controller is being serviced. It is set to 1 when the corresponding exception occurs. This bit does not affect acknowledging an exception request even when it is set to 1. 0: An exception other than an interrupt is not being serviced. 1: An exception other than an interrupt is being serviced. | R/W | 0 |
| 5 | ID | This bit disables the acknowledgement of EI level exception. When an EI level or FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level exception. As for the exceptions which the ID bit disables the acknowledgment, see the <i>Exception Cause List of RH850G3M User's Manual: Software</i> . This bit is also used to disable EI level exceptions from being acknowledged as a critical section while an ordinary program or interrupt is being serviced. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. The change of the ID bit by the EI or ID instruction will be enabled from the next instruction. 0: The acknowledgement of EI level exception is enabled. 1: The acknowledgement of EI level exception is disabled. | R/W | 1 |
| 4 | SAT* ¹ | This bit indicates that the operation result is saturated because the result of a saturated operation instruction operation has overflowed. This is a cumulative flag, so when the operation result of the saturated operation instruction becomes saturated, this bit is set to 1, but it is not cleared to 0 when the operation result for a subsequent instruction is not saturated. This bit is cleared to 0 by the LDSR instruction. This bit is neither set to 1 nor cleared to 0 when an arithmetic operation instruction is executed. 0: Not saturated 1: Saturated | R/W | 0 |
| 3 | CY | This bit indicates whether a carry or borrow has occurred in the operation result. 0: Carry or borrow has not occurred. 1: Carry or borrow has occurred. | R/W | 0 |
| 2 | OV* ¹ | This bit indicates whether or not an overflow has occurred during an operation. 0: Overflow has not occurred. 1: Overflow has occurred. | R/W | 0 |
| 1 | S* ¹ | This bit indicates whether or not the result of an operation is negative. 0: Result of operation is positive or 0. 1: Result of operation is negative | R/W | 0 |
| 0 | Z | This bit indicates whether or not the result of an operation is 0. 0: Result of operation is not 0. 1: Result of operation is 0. | R/W | 0 |

Note 1. The operation result of the saturation processing is determined in accordance with the contents of the OV flag and S flag during a saturated operation. When only the OV flag is set to 1 during a saturated operation, the SAT flag is set to 1.

| Operation result status | Flag status | | | Operation result after saturation processing |
|---------------------------------------|---------------------------------------|----|---|--|
| | SAT | OV | S | |
| Exceeded positive maximum value | 1 | 1 | 0 | 7FFF FFFF _H |
| Exceeded negative maximum value | 1 | 1 | 1 | 8000 0000 _H |
| Positive (maximum value not exceeded) | Value prior to operation is retained. | 0 | 0 | Operation result itself |
| Negative (maximum value not exceeded) | | | 1 | |

(f) EIIC — EI level exception source register

The EIIC register retains the source of any EI level exception that occurs. The value retained in this register is an exception code corresponding to a specific exception source.

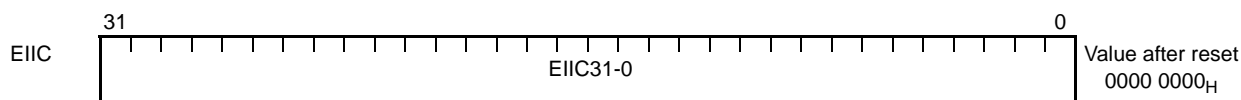


Table 3.11 EIIC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 0 | EIIC31-0 | These bits store the exception source code when an EI level exception is acknowledged. See the Table 6.11, Interrupt Exception Handler and Priority , and "Exception Cause List" in the <i>RH850G3M User's Manual: Software</i> about stored exception source code. The EIIC31-16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition, these bits are set to 0. | R/W | 0 |

(g) FEIC — FE level exception source register

The FEIC register retains the source of any FE level exception that occurs. The value retained in this register is an exception code corresponding to a specific exception source.

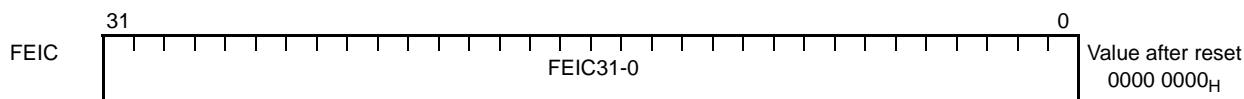


Table 3.12 FEIC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 0 | FEIC31-0 | These bits store the exception source code when an FE level exception is acknowledged. See the Table 6.11, Interrupt Exception Handler and Priority , and "Exception Cause List" in the <i>RH850G3M User's Manual: Software</i> about stored exception source code. The FEIC31-16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition, these bits are set to 0. | R/W | 0 |

(h) CTPC — Status save register when executing CALLT register

When a CALLT instruction is executed, the address of the next instruction after the CALLT instruction is saved to CTPC. Be sure to set an even-numbered address to the CTPC register. An odd-numbered address must not be specified.

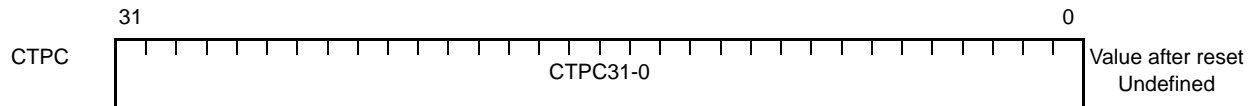


Table 3.13 CTPC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 1 | CTPC31-1 | These bits indicate the PC of the instruction after the CALLT instruction. | R/W | Undefined |
| 0 | CTPC0 | This bit indicates the PC of the instruction after the CALLT instruction. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the CTRET instruction is executed is 0. | R/W | Undefined |

(i) CTPSW — Status save register when executing CALLT register

When a CALLT instruction is executed, some of the PSW (program status word) settings are saved to CTPSW.

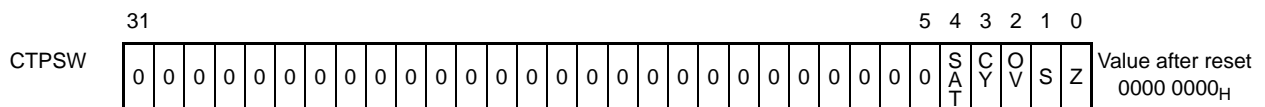


Table 3.14 CTPSW Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 5 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 4 | SAT | This bit stores the PSW.SAT bit setting when the CALLT instruction is executed. | R/W | 0 |
| 3 | CY | This bit stores the PSW.CY bit setting when the CALLT instruction is executed. | R/W | 0 |
| 2 | OV | This bit stores the PSW.OV bit setting when the CALLT instruction is executed. | R/W | 0 |
| 1 | S | This bit stores the PSW.S bit setting when the CALLT instruction is executed. | R/W | 0 |
| 0 | Z | This bit stores the PSW.Z bit setting when the CALLT instruction is executed. | R/W | 0 |

(j) CTBP — CALLT base pointer register

The CTBP register is used to specify table addresses of the CALLT instruction and generate target addresses.

Be sure to set the CTBP register to a Half-word address.

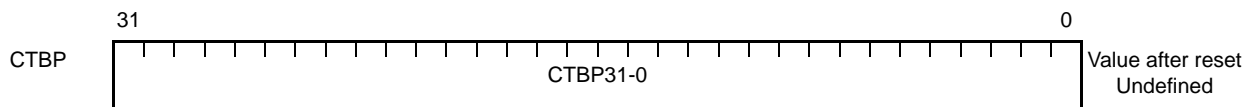


Table 3.15 CTBP Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 1 | CTBP31-1 | These bits indicate the base pointer address of the CALLT instruction. These bits indicate the first address in the table used by the CALLT instruction. | R/W | Undefined |
| 0 | CTBP0 | These bits indicate the base pointer address of the CALLT instruction. These bits indicate the first address in the table used by the CALLT instruction. Always set this bit to 0. | R | 0 |

(k) ASID — Address space ID register

This is the address space ID. This is used to identify the address space provided by the memory management function.

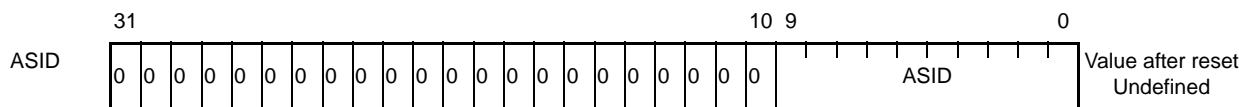


Table 3.16 ASID Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 10 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 9 to 0 | ASID | This is the address space ID. | R/W | Undefined |

(l) EIWR — EI level exception working register

The EIWR register is used as a working register when an EI level exception has occurred.

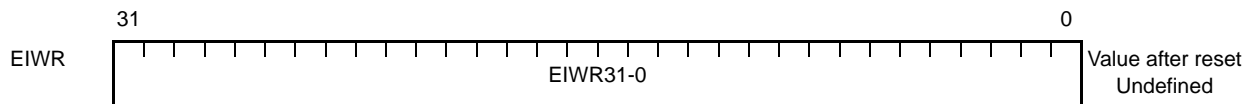


Table 3.17 EIWR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 0 | EIWR31-0 | These bits constitute a working register that can be used for any purpose during the processing of an EI level exception. Use this register for purposes such as storing the values of general-purpose registers. | R/W | Undefined |

(m) FEWR — FE level exception working register

The FEWR register is used as a working register when an FE level exception has occurred.

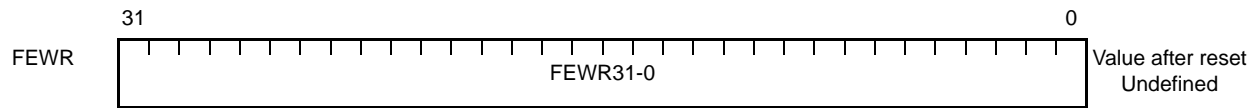


Table 3.18 FEWR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 0 | FEWR31-0 | These bits constitute a working register that can be used for any purpose during the processing of an FE level exception. Use this register for purposes such as storing the values of general-purpose registers. | R/W | Undefined |

(n) HTCFG0 — Thread configuration register

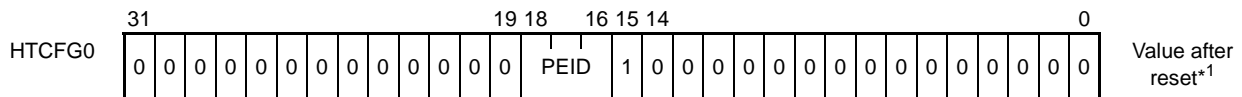


Table 3.19 HTCFG0 Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 19 | Reserved | (Reserved for future expansion. When read, the value after reset is read.) | R | 0 |
| 18 to 16 | PEID | These bits indicate the processor element number. | R | *2 |
| 15 | Reserved | (Reserved for future expansion. When read, the value after reset is read.) | R | 1 |
| 14 to 0 | Reserved | (Reserved for future expansion. When read, the value after reset is read.) | R | 0 |

Note 1. The value for CPU1 (PE1) in this device is 0001 8000_H.

Note 2. The value for CPU1 (PE1) in this device is 001_B.

(o) MEA — Memory error address register

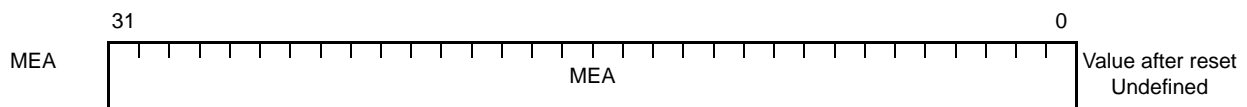


Table 3.20 MEA Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 0 | MEA | These bits store the violation address when an MAE (misaligned) or MPU occurs. | R/W | Undefined |

(p) MEI — Memory error information register

This register is used to store information about the instruction that caused the exception when a misaligned (MAE) or memory protection (MDP) exception occurs.

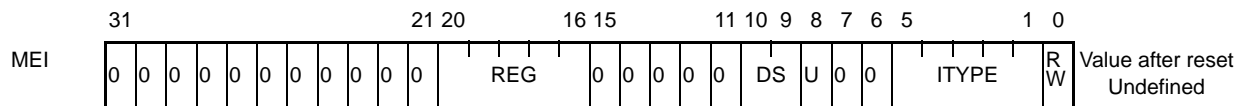


Table 3.21 MEI Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 21 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 20 to 16 | REG4-0 | These bits indicate the number of the source or destination register of the instruction that caused the exception. For details, see Table 3.22, Instructions Causing Exceptions and Values of MEI Register. | R/W | Undefined |
| 15 to 11 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 10, 9 | DS | These bits indicate the data type of the instruction that caused the exception.*1 0: Byte (8 bits) 1: Half-word (16 bits) 2: Word (32 bits) 3: Double-word (64 bits) For details, see Table 3.22, Instructions Causing Exceptions and Values of MEI Register. | R/W | Undefined |
| 8 | U | These bits indicate the sign extension method of the instruction that caused the exception. 0: Signed 1: Unsigned For details, see Table 3.22, Instructions Causing Exceptions and Values of MEI Register. | R/W | Undefined |
| 7, 6 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 5 to 1 | ITYPE4-0 | These bits indicate the instruction that caused the exception. For details, see Table 3.22, Instructions Causing Exceptions and Values of MEI Register. | R/W | Undefined |
| 0 | RW | This bit indicates whether the operation of the instruction that caused the exception was read (Load-memory) or write (Store-memory) 0: Read (Load-memory) 1: Write (Store-memory) For details, see Table 3.22, Instructions Causing Exceptions and Values of MEI Register. | R/W | Undefined |

Note 1. Even if the data is divided and access is made several times due to the specifications of the hardware, the original data type indicated by the instruction is stored.

Table 3.22 Instructions Causing Exceptions and Values of MEI Register (1/2)

| Instruction | REG | DS | U | RW | ITYPE |
|---------------|-----|---------------|--------------|-----------|--------------------|
| SLD.B | dst | 0 (Byte) | 0 (Signed) | 0 (Read) | 00000 _B |
| SLD.BU | dst | 0 (Byte) | 1 (Unsigned) | 0 (Read) | 00000 _B |
| SLD.H | dst | 1 (Half-word) | 0 (Signed) | 0 (Read) | 00000 _B |
| SLD.HU | dst | 1 (Half-word) | 1 (Unsigned) | 0 (Read) | 00000 _B |
| SLD.W | dst | 2 (Word) | 0 (Signed) | 0 (Read) | 00000 _B |
| SST.B | src | 0 (Byte) | 0 (Signed) | 1 (Write) | 00000 _B |
| SST.H | src | 1 (Half-word) | 0 (Signed) | 1 (Write) | 00000 _B |
| SST.W | src | 2 (Word) | 0 (Signed) | 1 (Write) | 00000 _B |
| LD.B (disp16) | dst | 0 (Byte) | 0 (Signed) | 0 (Read) | 00001 _B |

Table 3.22 Instructions Causing Exceptions and Values of MEI Register (2/2)

| Instruction | REG | DS | U | RW | ITYPE |
|---|-----|-----------------|--------------|------------------------|--------------------|
| LD.BU (disp16) | dst | 0 (Byte) | 1 (Unsigned) | 0 (Read) | 00001 _B |
| LD.H (disp16) | dst | 1 (Half-word) | 0 (Signed) | 0 (Read) | 00001 _B |
| LD.HU (disp16) | dst | 1 (Half-word) | 1 (Unsigned) | 0 (Read) | 00001 _B |
| LD.W (disp16) | dst | 2 (Word) | 0 (Signed) | 0 (Read) | 00001 _B |
| ST.B (disp16) | src | 0 (Byte) | 0 (Signed) | 1 (Write) | 00001 _B |
| ST.H (disp16) | src | 1 (Half-word) | 0 (Signed) | 1 (Write) | 00001 _B |
| ST.W (disp16) | src | 2 (Word) | 0 (Signed) | 1 (Write) | 00001 _B |
| LD.B (disp23) | dst | 0 (Byte) | 0 (Signed) | 0 (Read) | 00010 _B |
| LD.BU (disp23) | dst | 0 (Byte) | 1 (Unsigned) | 0 (Read) | 00010 _B |
| LD.H (disp23) | dst | 1 (Half-word) | 0 (Signed) | 0 (Read) | 00010 _B |
| LD.HU (disp23) | dst | 1 (Half-word) | 1 (Unsigned) | 0 (Read) | 00010 _B |
| LD.W (disp23) | dst | 2 (Word) | 0 (Signed) | 0 (Read) | 00010 _B |
| ST.B (disp23) | src | 0 (Byte) | 0 (Signed) | 1 (Write) | 00010 _B |
| ST.H (disp23) | src | 1 (Half-word) | 0 (Signed) | 1 (Write) | 00010 _B |
| ST.W (disp23) | src | 2 (Word) | 0 (Signed) | 1 (Write) | 00010 _B |
| LD.DW (disp23) | dst | 3 (Double-word) | 0 (Signed) | 0 (Read) | 00010 _B |
| ST.DW (disp23) | src | 3 (Double-word) | 0 (Signed) | 1 (Write) | 00010 _B |
| LDL.W | dst | 2 (Word) | 0 (Signed) | 0 (Read) | 00111 _B |
| STC.W | src | 2 (Word) | 0 (Signed) | 1 (Write) | 00111 _B |
| CAXI | dst | 2 (Word) | 1 (Unsigned) | 0 (Read) ^{*1} | 01000 _B |
| SET1 | — | 0 (Byte) | 1 (Unsigned) | 0 (Read) ^{*1} | 01001 _B |
| CLR1 | — | 0 (Byte) | 1 (Unsigned) | 0 (Read) ^{*1} | 01001 _B |
| NOT1 | — | 0 (Byte) | 1 (Unsigned) | 0 (Read) ^{*1} | 01001 _B |
| TST1 | — | 0 (Byte) | 1 (Unsigned) | 0 (Read) | 01001 _B |
| PREPARE | — | 2 (Word) | 1 (Unsigned) | 1 (Write) | 01100 _B |
| DISPOSE | — | 2 (Word) | 1 (Unsigned) | 0 (Read) | 01100 _B |
| PUSHSP | — | 2 (Word) | 1 (Unsigned) | 1 (Write) | 01101 _B |
| POPSP | — | 2 (Word) | 1 (Unsigned) | 0 (Read) | 01101 _B |
| SWITCH | — | 1 (Half-word) | 0 (Signed) | 0 (Read) | 10000 _B |
| CALLT | — | 1 (Half-word) | 1 (Unsigned) | 0 (Read) | 10001 _B |
| SYSCALL | — | 2 (Word) | 1 (Unsigned) | 0 (Read) | 10010 _B |
| CACHE | — | — | — | 0/1 ^{*2} | 10100 _B |
| Interrupt (table reference) ^{*3} | — | 2 (Word) | 1 (Unsigned) | 0 (Read) | 10101 _B |

Note 1. This exception occurs when the instruction executes a read access.

Note 2. It depends on actual operation.

Note 3. When the table reference interrupt vector is read.

NOTE

dst: Destination register number, src: Source register number

(q) RBASE — Reset vector base address register

This register indicates the reset vector address when there is a reset. If the PSW.EBV bit is 0, this vector address is also used as the exception vector address.

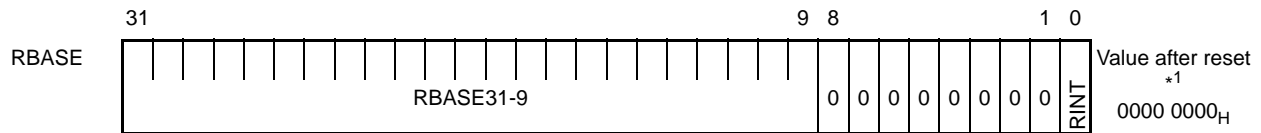


Table 3.23 RBASE Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|-----------|---|-----|-------------------|
| 31 to 9 | RBASE31-9 | These bits indicate the value of the upper 23 bits (bits 31 to 9) of the reset vector address when there is a reset. The lower 9 bits (bits 8 to 0) of the reset vector address are all handled as 0. This address is also used as the exception vector when PSW.EBV = 0. | R | *1 |
| 8 to 1 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 0 | RINT | When the RINT bit is set, the exception handler address for interrupt processing is reduced. For details, see "Direct vector method" in the <i>RH850G3M User's Manual: Software</i> . This bit is valid when PSW.EBV = 0. | R | 0 |

Note 1. Please refer to **Figure 35.4, Utilizing the Variable Reset Vector Function to Update the Boot Program** for detail.

(r) EBASE — Exception handler vector address register

This register indicates the exception handler vector address. This register is valid when the PSW.EBV bit is 1.

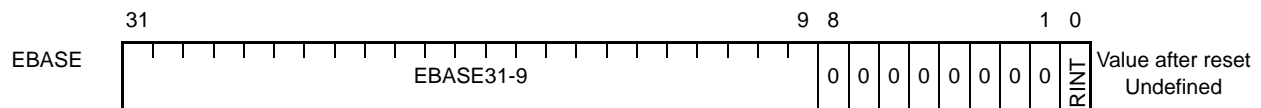


Table 3.24 EBASE Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|-----------|---|-----|-------------------|
| 31 to 9 | EBASE31-9 | The upper 23 bits (bits 31 to 9) of the base address of the exception handler routine are set to the address specified by this register. The lower 9 bits (bits 8 to 0) of the base address are all handled as 0, and are converted to an address to which the offset address of each exception is added. | R/W | Undefined |
| 8 to 1 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 0 | RINT | When the RINT bit is set, the offset address of the exception handler used during interrupt servicing is decremented. For details, see "Direct vector method" in the <i>RH850G3M User's Manual: Software</i> . | R/W | Undefined |

(s) INTBP — Base address of the interrupt handler table register

This register indicates the base address of the address table when the table reference method is selected as the interrupt handler address selection method.

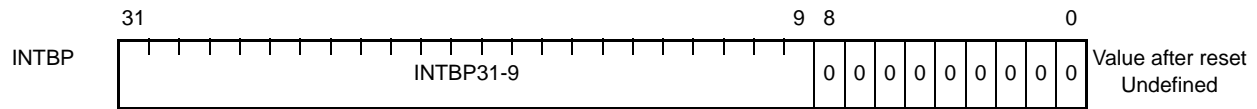


Table 3.25 INTBP Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|-----------|--|-----|-------------------|
| 31 to 9 | INTBP31-9 | These bits indicate the base pointer address of the table reference method of the interrupt. The value indicated by these bits is the first address in the table used to determine the exception handler when the interrupt specified to use the table reference method (EIINT0 to EIINT383) is acknowledged. The bit names are not assigned because EBASE8-0 bits are always 0. | R/W | Undefined |
| 8 to 0 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |

(t) PID — Processor ID register

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.

CAUTION

The PID register indicates information used to identify the incorporated CPU core and CPU core configuration. Usage such that the software behavior varies dynamically according to the PID register information is not assumed.

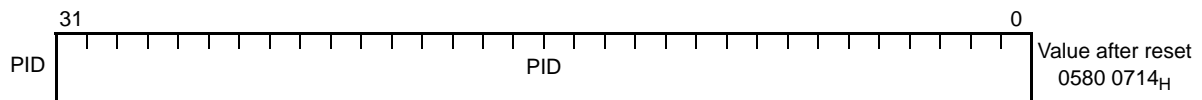


Table 3.26 PID Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 24 | PID | Architecture identifier This identifier indicates the architecture of the processor. | R | 05 _H |
| 23 to 8 | | Function identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bit 23 to 11 Reserved Bit 10 Double-precision floating-point operation function Bit 9 Single-precision floating-point operation function Bit 8 Memory protection function (MPU) | R | 8007 _H |
| 7 to 0 | | Version identifier This identifier indicates the version of the processor. | R | 14 _H |

(u) SCCFG — SYSCALL operation setting register

This register is used to specify operations related to the SYSCALL instruction. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

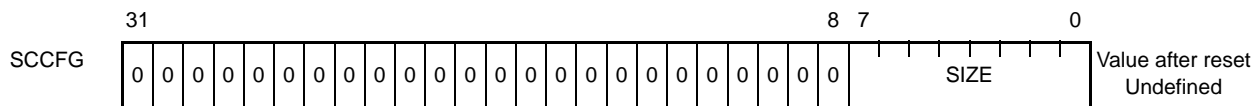


Table 3.27 SCCFG Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 8 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 7 to 0 | SIZE | These bits specify the maximum number of entries of a table that the SYSCALL instruction references. The maximum number of entries the SYSCALL instruction references is 1 if SIZE is 0, and 256 if SIZE is 255. By setting the maximum number of entries appropriately in accordance with the number of functions branched by the SYSCALL instruction, the memory area can be effectively used. If a vector exceeding the maximum number of entries is specified for the SYSCALL instruction, the first entry is selected. Place an error processing routine at the first entry. | R/W | Undefined |

(v) SCBP — SYSCALL base pointer register

The SCBP register is used to specify a table address of the SYSCALL instruction and generate a target address. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

Be sure to set a word address to the SCBP register.

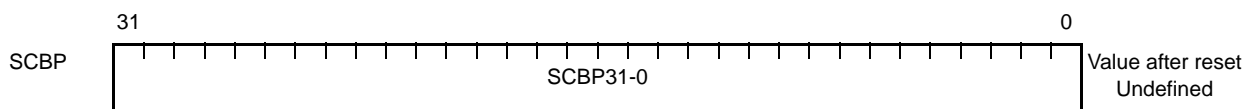


Table 3.28 SCBP Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 2 | SCBP31-2 | These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the first address in the table used by the SYSCALL instruction. | R/W | Undefined |
| 1, 0 | SCBP1-0 | These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the first address in the table used by the SYSCALL instruction. Always set these bits to 0. | R | 0 |

(w) MCFG0 — Machine configuration register

This register indicates the CPU configuration.

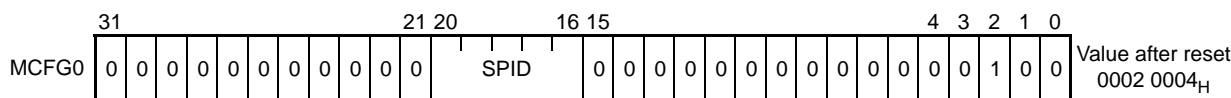


Table 3.29 MCFG0 Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|--------------------|
| 31 to 21 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 20 to 16 | SPID | These bits indicate the system protection number. It is not possible to write "00000 _B "/"00001 _B " to SPID. If written, these value is "00010 _B ". | R/W | 00010 _B |
| 15 to 4 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 3 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 2 | Reserved | (Reserved for future expansion. Be sure to set to 1.) | R | 1 |
| 1 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 0 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |

(x) MCTL — Machine control register

This register is used to control the CPU.

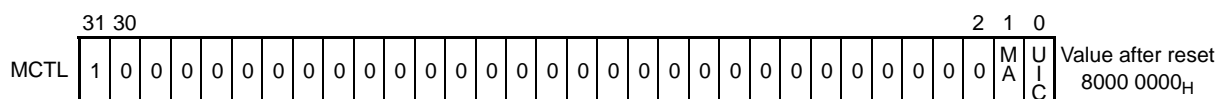


Table 3.30 MCTL Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 | Reserved | (Reserved for future expansion. Be sure to set to 1.) | R | 1 |
| 30 to 2 | Reserved | (Reserved for future expansion. Be sure to set to 0.) | R | 0 |
| 1 | MA | This bit specifies the operation when a misaligned access occurs. 0: An exception occurs.*1 1: An exception does not occur.*2 | R/W | 0 |
| 0 | UIC | This bit is used to control the interrupt enable/disable operation in user mode. When this bit is set to 1, executing the EI/DI instruction become possible in user mode. | R/W | 0 |

Note 1. Excluding LD.DW and ST.DW instructions executed at an address at a word boundary.

Note 2. If the MA bit is 1, the CPU divides a single misaligned data access into multiple aligned accesses. In this case, the instruction atomicity is not guaranteed.

When MA bit is “1”, CPU divides the transaction from one misaligned data access to multiple aligned data accesses. In this case, atomic characteristics of the instruction are not guaranteed.

| Access Conditions | | Cycles Divided by the CPU | | |
|-------------------|---------|---------------------------|-------------------------|------------------------|
| Data Width | Address | 1st | 2nd | 3rd |
| 16 bits | 2n + 1 | 8-bit access to 2n + 1 | 8-bit access to 2n + 2 | — |
| 32 bits | 4n + 1 | 8-bit access to 4n + 1 | 16-bit access to 4n + 2 | 8-bit access to 4n + 4 |
| 32 bits | 4n + 2 | 16-bit access to 4n + 2 | 16-bit access to 4n + 4 | — |
| 32 bits | 4n + 3 | 8-bit access to 4n + 3 | 16-bit access to 4n + 4 | 8-bit access to 4n + 6 |

(3) Interrupt Function Registers

(a) Interrupt function system register

Reading from/writing to an interrupt function system register is performed by specifying the system register number, which consists of the register number and selection identifier, with the LDSR or STSR instructions.

Table 3.31 Interrupt Function System Registers

| Register No. (reg.ID, sel.ID) | Symbol | Function | Access Permission |
|----------------------------------|--------|--|-------------------|
| SR7, 1 | FPIPR | FPI exception interrupt priority setting | SV |
| SR10, 2 | ISPR | Priority of interrupt being serviced | SV |
| SR11, 2 | PMR | Interrupt priority masking | SV |
| SR12, 2 | ICSR | Interrupt control status | SV |
| SR13, 2 | INTCFG | Interrupt function setting | SV |

(b) FPIPR — FPI exception interrupt priority setting register

This register specifies the FPI exception interrupt priority.

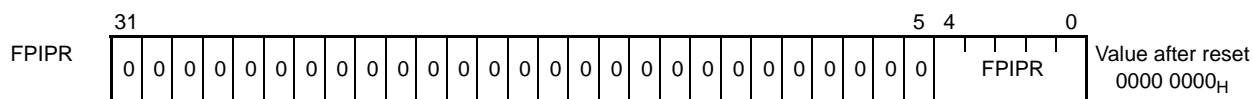


Table 3.32 FPIPR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 5 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 4 to 0 | FPIPR | These bits specify the priority of the floating-point operation exception interrupt (FPI, indicating imprecision). Values from 0 to 16 should be used; a value greater than or equal to 17 is prohibited. FPI exceptions are handled according to this interrupt priority, which is specified in advance. When generated at the same time as another interrupt with the same priority level, the FPI takes priority. | R/W | 0 |

NOTE

If these bits are set to 16 or higher values, handling is as if the setting were 16.

(c) ISPR — Priority of interrupt being serviced register

This register holds the priority of the EIINTn interrupt being serviced by the CPU per priority. This priority value is then used to perform priority ceiling processing when multiple interrupts are generated.

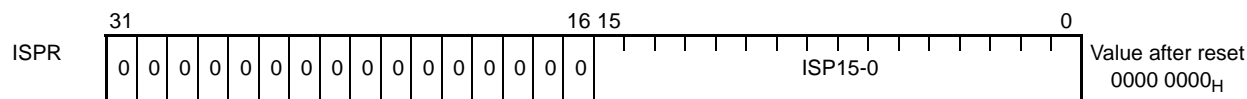


Table 3.33 ISPR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----------------|-------------------|
| 31 to 16 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 15 to 0 | ISP15-0 | These bits indicate the acknowledgment status of an EIINTn interrupt with a priority that corresponds to the relevant bit position* ¹ . 0: An interrupt request for an interrupt whose priority corresponds to the relevant bit position has not been acknowledged. 1: An interrupt request for an interrupt whose priority corresponds to the relevant position is being serviced by the CPU core. | R* ⁴ | 0 |

The bit positions correspond to the following priority levels:

| Bit | Priority |
|-----|----------------------|
| 0 | Priority 0 (highest) |
| 1 | Priority 1 |
| : | : |
| 14 | Priority 14 |
| 15 | Priority 15 |

When an interrupt request (EIINTn) is acknowledged, the bit corresponding to the acknowledged interrupt request is automatically set to 1. If PSW.EP is 0 when the EIRET instruction is executed, the bit with the highest priority among the ISP15-0 bits that are set (0 is the highest priority) is cleared to 0.*²

While a bit in this register is set to 1, same and lower priority interrupts (EIINTn) and FPI exception*³ are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged.

When performing software-based priority control using the PMR register, be sure to clear this register by using the INTCFG.ISPC bit functions.

- Note 1. For details, see “Exception Priority of Interrupts and the Priority Mask” in the *RH850G3M User’s Manual: Software*.
- Note 2. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting (1) the INTCFG.ISPC bit. It is recommended to enable auto-updating of values, so in normal cases, the INTCFG.ISPC bit should be cleared to 0.
- Note 3. Since FPI exceptions have the same level of priority as this interrupt (EIINTn), they are affected by the ISPR in the same way as the interrupt. The priority of the FPI exception is set by the FPIPR register.
- Note 4. This is R or R/W, depending on the setting of the INTCFG.ISPC bit. It is recommended to use this register as a read-only (R) register in normal cases.

(d) PMR — Interrupt priority masking register

This register is used to mask the specified interrupt priority.

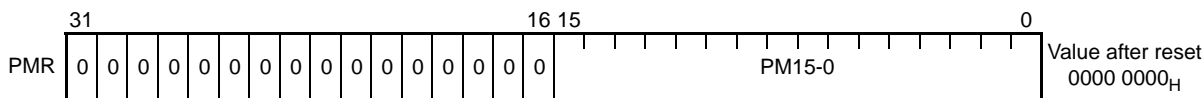


Table 3.34 PMR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 16 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 15 to 0 | PM15-0 | These bits mask an interrupt request with a priority level that corresponds to the relevant bit position. 0: Servicing of an interrupt request with a priority that corresponds to the relevant bit position is enabled. 1: Servicing of an interrupt request with a priority that corresponds to the relevant bit position is disabled. | R/W | 0 |

The bit positions correspond to the following priority levels:

| Bit | Priority |
|-----|--------------------------------------|
| 0 | Priority 0 (highest) |
| 1 | Priority 1 |
| : | : |
| 14 | Priority 14 |
| 15 | Priority 15 and priority 16 (lowest) |

While a bit in this register is set to 1, interrupts (EIINTn) with the priority corresponding to that bit and FPI exception*¹ are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged*².

- Note 1. Since a FPI exception is specified as the same level of priority as that of an interrupt (EIINTn), it is affected by the PMR like interrupts. The priority of FPI exception is set by the FPIPR register.
- Note 2. Specify the masks by setting the bits to 1 in order from the lowest-priority bit. For example, FF00_H can be set, but F0F0_H or 00FF_H cannot.

(e) ICSR — Interrupt control status register

This register indicates the interrupt control status in the CPU.

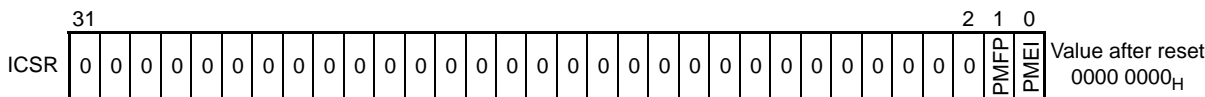


Table 3.35 ICSR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 2 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 1 | PMFP | This bit indicates that an FPI exception with the priority level masked by the PMR register exists. | R | 0 |
| 0 | PMEI | This bit indicates that an interrupt (EIINTn) with the priority level masked by the PMR register exists | R | 0 |

(4) FPU Function Registers

(a) Floating-point operation registers

The FPU uses the CPU general-purpose registers (r0 to r31). There are no register files used only for floating-point operations.

- Single-precision floating-point instruction

Thirty-two 32-bit registers can be specified. These correspond to the general-purpose registers r0 to r31.

- Double-precision floating-point instruction

Sixteen 64-bit registers can be specified. These correspond to the register pairs each using one pair of general-purpose registers, ({r1, r0}, {r3, r2} ... {r31, r30}). Each register pair is specified in the instruction format with an even numbered register. Because r0 is a zero register (always holds "0"), in principle {r1, r0} should not be used by a double-precision floating-point instruction.

(b) Floating-point system registers

The FPU can use the following system registers to control floating-point operation. Reading from/writing to a floating-point system register is performed by specifying the system register number, which consists of the register number and selection identifier, with the LDSR or STSR instructions. For details of the registers, see "*Floating-Point Function System Registers*" in the *RH850G3M User's Manual: Software*.

Table 3.37 FPU System Registers

| Register No. (reg.ID, sel.ID) | Symbol | Function | Access Permission |
|----------------------------------|--------|--|----------------------|
| SR6, 0 | FPSR | Floating-point operation setting/status | CU0 and SV |
| SR7, 0 | FPEPC | Floating-point operation exception program counter | CU0 and SV |
| SR8, 0 | FPST | Floating-point status | CU0 |
| SR9, 0 | FPCCR | Floating-point operation comparison result | CU0 |
| SR10, 0 | FPCFG | Floating-point function setting | CU0 |
| SR11, 0 | FPEC | Floating-point operation exception control | CU0 and SV |

(5) MPU Function Registers**(a) MPU function system registers**

Reading from/writing to an MPU function system register is performed by specifying the system register number, which consists of the register number and selection identifier, with the LDSR or STSR instructions.

Table 3.38 MPU Function System Registers (1/2)

| Register No. (reg.ID, sel.ID) | Symbol | Function | Access Permission |
|----------------------------------|--------|--|----------------------|
| SR0, 5 | MPM | Memory protection operation mode setting | SV |
| SR1, 5 | MPRC | MPU region control | SV |
| SR4, 5 | MPBRGN | MPU base region number | SV |
| SR5, 5 | MPTRGN | MPU end region number | SV |
| SR8, 5 | MCA | Memory protection setting check address | SV |
| SR9, 5 | MCS | Memory protection setting check size | SV |
| SR10, 5 | MCC | Memory protection setting check command | SV |
| SR11, 5 | MCR | Memory protection setting check result | SV |
| SR0, 6 | MPLA0 | Protection area minimum address | SV |
| SR1, 6 | MPUA0 | Protection area maximum address | SV |
| SR2, 6 | MPAT0 | Protection area attribute | SV |
| SR4, 6 | MPLA1 | Protection area minimum address | SV |
| SR5, 6 | MPUA1 | Protection area maximum address | SV |
| SR6, 6 | MPAT1 | Protection area attribute | SV |
| SR8, 6 | MPLA2 | Lower address of the protection area | SV |
| SR9, 6 | MPUA2 | Protection area maximum address | SV |
| SR10, 6 | MPAT2 | Protection area attribute | SV |
| SR12, 6 | MPLA3 | Protection area minimum address | SV |
| SR13, 6 | MPUA3 | Protection area maximum address | SV |
| SR14, 6 | MPAT3 | Protection area attribute | SV |
| SR16, 6 | MPLA4 | Protection area minimum address | SV |
| SR17, 6 | MPUA4 | Protection area maximum address | SV |
| SR18, 6 | MPAT4 | Protection area attribute | SV |
| SR20, 6 | MPLA5 | Protection area minimum address | SV |
| SR21, 6 | MPUA5 | Protection area maximum address | SV |
| SR22, 6 | MPAT5 | Protection area attribute | SV |
| SR24, 6 | MPLA6 | Protection area minimum address | SV |
| SR25, 6 | MPUA6 | Protection area maximum address | SV |
| SR26, 6 | MPAT6 | Protection area attribute | SV |
| SR28, 6 | MPLA7 | Protection area minimum address | SV |
| SR29, 6 | MPUA7 | Protection area maximum address | SV |
| SR30, 6 | MPAT7 | Protection area attribute | SV |
| SR0, 7 | MPLA8 | Protection area minimum address | SV |
| SR1, 7 | MPUA8 | Protection area maximum address | SV |
| SR2, 7 | MPAT8 | Protection area attribute | SV |
| SR4, 7 | MPLA9 | Protection area minimum address | SV |
| SR5, 7 | MPUA9 | Protection area maximum address | SV |

Table 3.38 MPU Function System Registers (2/2)

| Register No. (reg.ID, sel.ID) | Symbol | Function | Access Permission |
|----------------------------------|--------|---------------------------------|-------------------|
| SR6, 7 | MPAT9 | Protection area attribute | SV |
| SR8, 7 | MPLA10 | Protection area minimum address | SV |
| SR9, 7 | MPUA10 | Protection area maximum address | SV |
| SR10, 7 | MPAT10 | Protection area attribute | SV |
| SR12, 7 | MPLA11 | Protection area minimum address | SV |
| SR13, 7 | MPUA11 | Protection area maximum address | SV |
| SR14, 7 | MPAT11 | Protection area attribute | SV |
| SR16, 7 | MPLA12 | Protection area minimum address | SV |
| SR17, 7 | MPUA12 | Protection area maximum address | SV |
| SR18, 7 | MPAT12 | Protection area attribute | SV |
| SR20, 7 | MPLA13 | Protection area minimum address | SV |
| SR21, 7 | MPUA13 | Protection area maximum address | SV |
| SR22, 7 | MPAT13 | Protection area attribute | SV |
| SR24, 7 | MPLA14 | Protection area minimum address | SV |
| SR25, 7 | MPUA14 | Protection area maximum address | SV |
| SR26, 7 | MPAT14 | Protection area attribute | SV |
| SR28, 7 | MPLA15 | Protection area minimum address | SV |
| SR29, 7 | MPUA15 | Protection area maximum address | SV |
| SR30, 7 | MPAT15 | Protection area attribute | SV |

(b) MPM — Memory protection operation mode register

The memory protection mode register is used to define the basic operating state of the memory protection function.

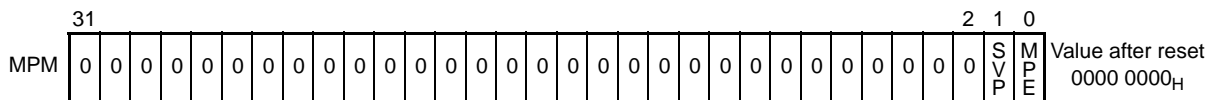


Table 3.39 MPM Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 2 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 1 | SVP | In SV mode. (when PSW.UM = 0), this bit is used to specify whether to restrict access according to the SX, SW, and SR bits of the MPAT register for each protection area.*1 0: As usual, implicitly enable all access in SV mode. 1: Restrict access according to the SX, SW, and SR bits even in SV mode.*2 | R/W | 0 |
| 0 | MPE | This bit is used to specify whether to enable or disable MPU function. 0: Disable 1: Enable | R/W | 0 |

Note 1. When the SVP bit is set to 1, access is restricted according to the setting of each protection area even in SV mode. Therefore, specify protection areas before setting the SVP bit to prevent the access of the program itself from being restricted.

Note 2. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access to the memory area necessary for the exception handler and exception handling is permitted.

(c) MPRC — MPU region control register

Bits used to perform special memory protection function operations are located in this register.

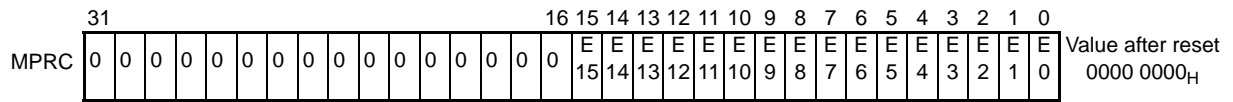


Table 3.40 MPRC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 16 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 15 to 0 | E15-E0 | These are the enable bits for each protection area. Bit En is a copy of bit MPATn.E (where n = 15 to 0). | R/W | 0 |

(d) MPBRGN — MPU base region register

This register indicates the minimum usable MPU area number.

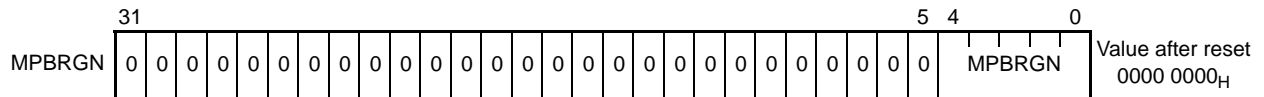


Table 3.41 MPBRGN Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 5 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 4 to 0 | MPBRGN | These bits indicate the smallest number of an MPU area. These bits are always read as 0. | R | 0 |

(e) MPTRGN — MPU end region register

This register indicates the maximum usable MPU area number + 1.

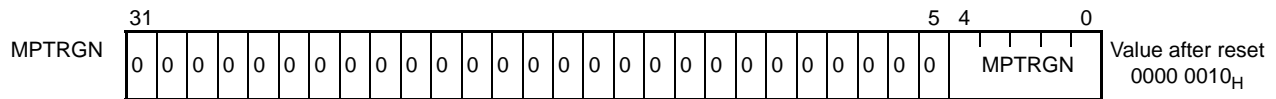


Table 3.42 MPTRGN Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 5 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 4 to 0 | MPTRGN | These bits indicate the largest number of an MPU area + 1. These bits indicate the maximum number of MPU areas incorporated into the hardware. The number of protection areas in this device is 16. | R | 10 _H |

(f) MCA — Memory protection setting check address register

This register is used to specify the base address of the area for which a memory protection setting check is to be performed.

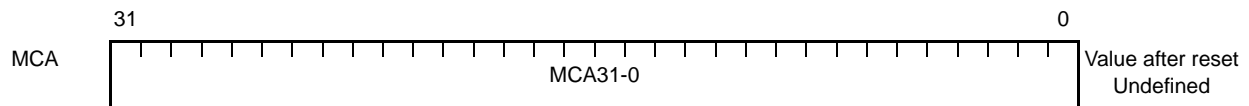


Table 3.43 MCA Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 0 | MCA31-0 | These bits are used to specify the start address of the memory area subject to a memory protection setting check in bytes. | R/W | Undefined |

(g) MCS — Memory protection setting check size register

This register is used to specify the size of the area for which a memory protection setting check is to be performed.

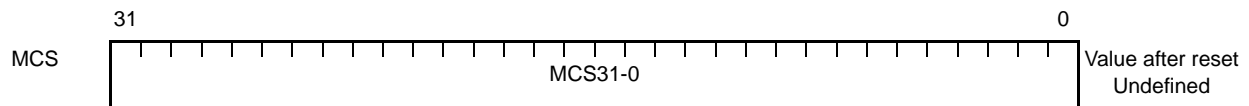


Table 3.44 MCS Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 0 | MCS31-0 | These bits specify the size of the target area to specify the size of the memory area subject to a memory protection setting check in bytes. Checking in areas below the address value in the MCS register is not possible because the specified size is handled as an unsigned integer. Do not set 0000 0000 _H in the MCS register. | R/W | Undefined |

(h) MCC — Memory protection setting check command register

This command register is used to start a memory protection setting check.

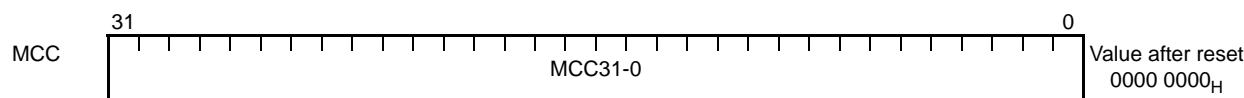


Table 3.45 MCC Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 0 | MCC31-0 | When any value is written to the MCC register, a memory protection setting check starts. Setting the MCA and MCS registers and then writing to this register leads to storage of the result of checking in the MCR register. Since writing any value to this register starts the check, doing so does not require any extra registers when r0 is used as a source register. The result of checking is reflected according to any area setting regardless of the setting of the PSW.UM bit. The value read from the MCC register is always 0000 0000 _H . | R/W | 0 |

(i) MCR — Memory protection setting check result register

This register is used to store the results of a memory protection setting check.

CAUTION

1. If the specified area to be checked crosses 0000 0000_H, it is judged as an area setting error, and the MCR.OV bit is set to 1. This means that the MCR.OV bit must be checked to access the check results. Do not use the check result until it is confirmed that the result is not invalid (OV = 0).
2. When the default set (MPM.DX, DW, DR) is set to 1, a correct result may not be obtained in some cases. If the specified default is to be enabled, do not use the memory protection setting check function.

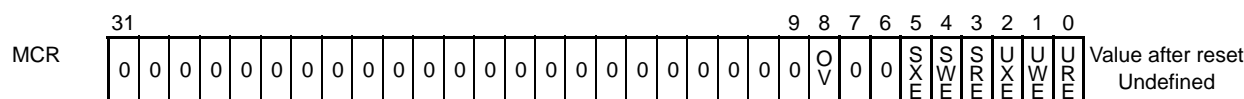


Table 3.46 MCR Register Contents (1/2)

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 9 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 8 | OV | If the specified area includes 0000 0000 _H or 7FFF FFFF _H , 1 is stored in this bit. In other cases, 0 is stored in this bit. | R/W | Undefined |
| 7, 6 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 5 | SXE | If the specified area is contained within one protection area and execution is permitted for that area in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit. | R/W | Undefined |
| 4 | SWE | If the specified area is contained within one protection area and writing to that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit. | R/W | Undefined |
| 3 | SRE | If the specified area is contained within one protection area and reading from that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit. | R/W | Undefined |

Table 3.46 MCR Register Contents (2/2)

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 2 | UXE | If the specified area is contained within one protection area and execution is permitted for that area in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit. | R/W | Undefined |
| 1 | UWE | If the specified area is contained within one protection area and writing to that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit. | R/W | Undefined |
| 0 | URE | If the specified area is contained within one protection area and reading from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit. | R/W | Undefined |

(j) MPLAn — Protection area minimum address register

These registers indicate the minimum address of area n (where n = 0 to 15).

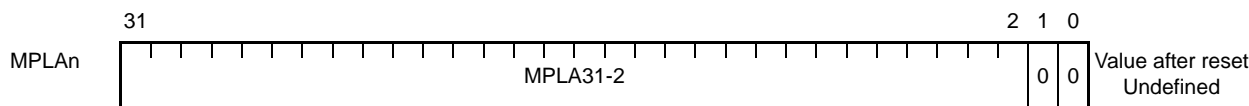


Table 3.47 MPLAn Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 2 | MPLA31-2 | These bits indicate the minimum address of area n. The MPLAn.MPLA1-0 bits are implicitly set to 0. | R/W | Undefined |
| 1, 0 | Reserved | Reserved for future expansion. Be sure to clear these bits to 0. | R | 0 |

(k) MPUAn — Protection area maximum address register

These registers indicate the maximum address of area n (where n = 0 to 15).

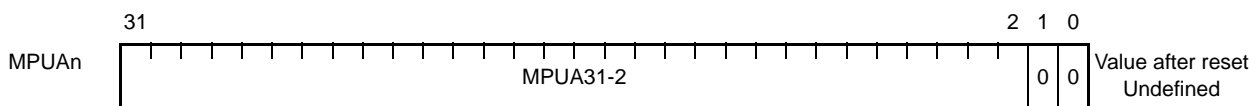


Table 3.48 MPUAn Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 2 | MPUA31-2 | These bits indicate the maximum address of area n. The MPUAn.MPUA1-0 bits are implicitly set to 1. | R/W | Undefined |
| 1, 0 | Reserved | Reserved for future expansion. Be sure to clear these bits to 0. | R | 0 |

(I) MPATn — Protection area attribute register

These registers indicate the attributes of area n (where n = 0 to 15).

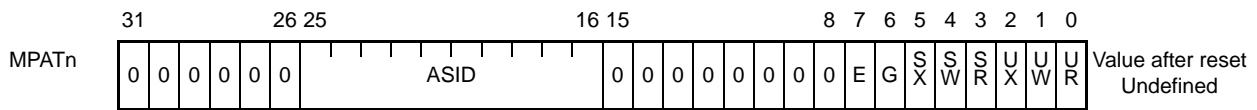


Table 3.49 MPATn Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 to 26 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 25 to 16 | ASID | These bits indicate the ASID value to be used as the area match condition. | R/W | Undefined |
| 15 to 8 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 7 | E | This bit indicates whether area n is enabled or disabled. 0: Area n is disabled. 1: Area n is enabled. | R/W | 0 |
| 6 | G | 0: ASID match is the condition. 1: ASID match is not the condition. When this bit is 0, the condition of the area match is MPATn.ASID = ASID.ASID. When this bit is 1, the values of MPATn.ASID and ASID.ASID is not the condition. | R/W | Undefined |
| 5 | SX | This bit indicates the execution privilege in supervisor mode.*1 0: Execution is disabled. 1: Execution is enabled. | R/W | Undefined |
| 4 | SW | This bit indicates the write permission in supervisor mode.*1 0: Writing is disabled. 1: Writing is enabled. | R/W | Undefined |
| 3 | SR | This bit indicates the read permission in supervisor mode*1 0: Reading is disabled. 1: Reading is enabled. | R/W | Undefined |
| 2 | UX | This bit indicates the execution privilege in user mode. 0: Execution is disabled. 1: Execution is enabled. | R/W | Undefined |
| 1 | UW | This bit indicates the write permission in user mode. 0: Writing is disabled. 1: Writing is enabled. | R/W | Undefined |
| 0 | UR | This bit indicates the read permission in user mode. 0: Reading is disabled. 1: Reading is enabled. | R/W | Undefined |

Note 1. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access to the memory area necessary for the exception handler and exception handling is permitted.

(6) Cache Operation Function Registers

(a) Cache control function system registers

Cache control function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.50 Cache Operation Function Registers

| Register No. (reg.ID, sel.ID) | Symbol | Function | Access Permission |
|----------------------------------|--------|----------------------------------|----------------------|
| SR16, 4 | ICTAGL | Instruction cache tag Lo access | SV |
| SR17, 4 | ICTAGH | Instruction cache tag Hi access | SV |
| SR18, 4 | ICDACL | Instruction cache data Lo access | SV |
| SR19, 4 | ICDATH | Instruction cache data Hi access | SV |
| SR24, 4 | ICCTRL | Instruction cache control | SV |
| SR26, 4 | ICCFG | Instruction cache configuration | SV |
| SR28, 4 | ICERR | Instruction cache error | SV |

(b) ICTAGL — Instruction cache tag Lo access register

This register is used for CIST and CILD instructions for the instruction cache. This register holds values to be stored in the tag RAM of the instruction cache by the execution of CIST instructions and values read from the tag RAM of the instruction cache by the execution of CILD instructions.

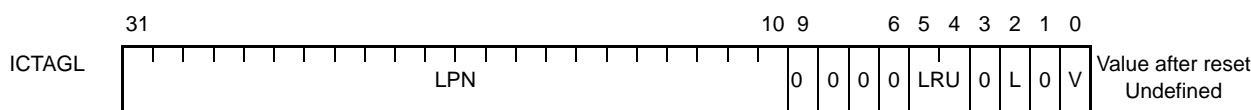


Table 3.51 ICTAGL Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|----------------------|
| 31 to 10 | LPN | These bits hold the values of 24 to 12 bits of physical page numbers. When writing, always write 0 to bits 31 to 25 and bits 11 to 10. | R/W | Undefined |
| 9 to 6 | Reserved | (Reserved for future expansion. When writing, always write 0 to these bits.) | R | 0 |
| 5, 4 | LRU | These bits indicate the LRU information of the specified cache line. The CIST instruction cannot be used to change the LRU information to desired values. | R/W | Undefined |
| 3 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 2 | L | This bit holds the lock information. | R/W | Undefined |
| 1 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 0 | V | This bit retains whether the specified cache line is enabled or disabled. | R/W | Undefined |

(c) ICTAGH — Instruction cache tag Hi access register

This register is used for a CIST or CILD instruction for the instruction cache. This register stores the value to be stored in the instruction cache tag RAM on CIST execution and the value read from the instruction cache tag RAM on CILD execution.

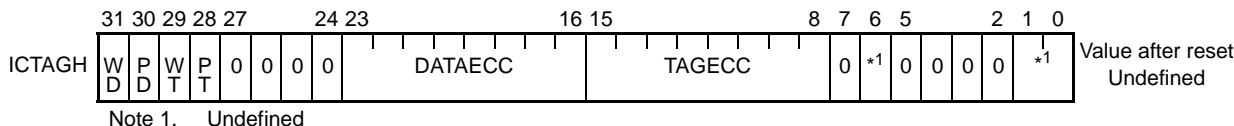


Table 3.52 ICTAGH Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|---|-----|-------------------|
| 31 | WD | When this bit is set to 1, executing a CIST instruction updates the cache data RAM. | R/W | Undefined |
| 30 | PD | When this bit is set to 1 during CIST execution, values in the DATAECC field are overwritten to ECC for data RAM. When this value is 0, ECC is generated automatically from the write data. | R/W | Undefined |
| 29 | WT | When this bit is set to 1 during CIST execution, tag RAM of cache is updated. | R/W | Undefined |
| 28 | PT | When this bit is set to 1 during CIST execution, values in the TAGECC field are overwritten to ECC for tag RAM. When this value is 0, ECC is generated automatically from the write data. | R/W | Undefined |
| 27 to 24 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 23 to 16 | DATAECC | These bits store ECC for data RAM. | R/W | Undefined |
| 15 to 8 | TAGECC | These bits store ECC for tag RAM. Write 0 to bit 15. | R/W | Undefined |
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | R | 0 |
| 6 | Reserved | When read, an undefined value is read. When writing, always write 0. | R | Undefined |
| 5 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | R | 0 |
| 1, 0 | Reserved | When read, an undefined value is read. When writing, always write 0. | R | Undefined |

(d) ICDATL — Instruction cache data Lo access register

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of CIST, values that are stored to the data RAM for the instruction cache are stored. During execution of CILD, values read from the data RAM for the instruction cache are stored.

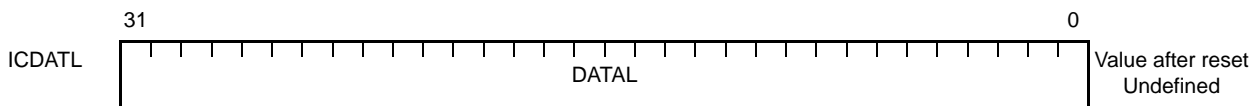


Table 3.53 ICDATL Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 0 | DATAL | Bits 31 to 0 or bits 95 to 64 are stored among the instruction data of a block in the specified cache line. The stored bits are specified by the offset of index. Offset of index = 0000: Bits 31 to 0 Offset of index = 1000: Bits 95 to 64 | R/W | Undefined |

(e) ICDATH — Instruction cache data Hi access register

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of CIST, values that are stored to the data RAM for the instruction cache are stored. During execution of CILD, values read from the data RAM for the instruction cache are stored.

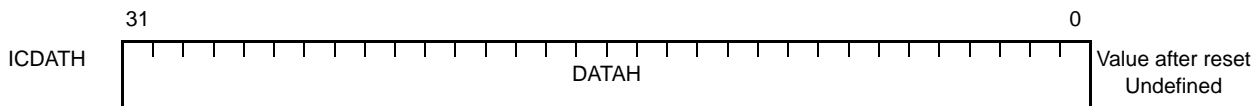


Table 3.54 ICDATH Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 0 | DATAH | Bits 63 to 32 or bits 127 to 96 are stored among the instruction data of a block in the specified cache line. The stored bits are specified by the offset of index. Offset of index = 0000: Bits 63 to 32 Offset of index = 1000: Bits 127 to 96 | R/W | Undefined |

(f) ICCTRL — Instruction cache control register

This register controls the instruction cache.

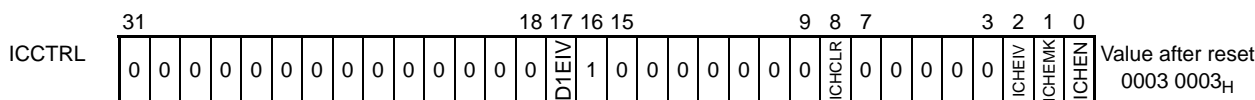


Table 3.55 ICCTRL Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 18 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 17 | D1EIV | This bit selects the operation in response to 1-bit errors in the data RAM. 0: The error is corrected and then processing continues, but the address of the entry that had an error is retained. 1: The error is not corrected, the entry is cleared, and fetching is repeated. This bit is read as the previous value until the setting is actually reflected in the instruction cache. | R/W | 1 |
| 16 | Reserved | (Reserved for future expansion. Be sure to set to 1.) | R | 1 |
| 15 to 9 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 8 | ICHCLR | Setting this bit to 1 clears the whole instruction cache in a single operation. After this bit has been set to 1, it will be read as 1 until clearing is completed. The read value of this bit is 0 once clearing of the cache is completed. | R/W | 0 |
| 7 to 3 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 2 | ICHEIV | Setting this bit to 1 allows the instruction cache to be automatically disabled (ICHEN bit to be cleared to 0) when a cache error occurs. | R/W | 0 |
| 1 | ICHEMK | Setting this bit to 1 selects masking of cache error exception notifications for the CPU when cache errors are encountered. | R/W | 1 |
| 0 | ICHEN | This bit disables or enables the instruction cache. 0: Instruction cache is disabled. 1: Instruction cache is enabled. This bit is read as the previous value until the setting is actually reflected in the instruction cache. | R/W | 1 |

(g) ICCFG — Instruction cache configuration register

This register shows the configuration of the instruction cache.

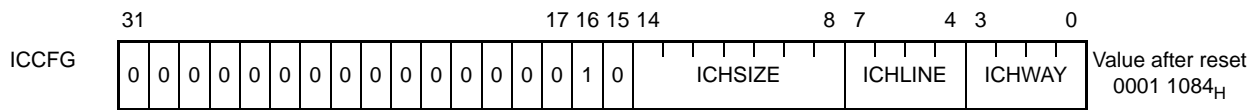


Table 3.56 ICCFG Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 17 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 16 | Reserved | (Reserved for future expansion. Be sure to set to 1.) | R | 1 |
| 15 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 14 to 8 | ICHSIZE | These bits indicate the capacity (in Kbytes) of the instruction cache. 001 0000: 16 Kbytes | R | 10 _H |
| 7 to 4 | ICHLINE | These bits indicate the number of lines per 1 way in the instruction cache. 1000: 256 lines | R | 8 _H |
| 3 to 0 | ICHWAY | These bits indicate the number of ways of the instruction cache. 0100: 4 ways | R | 4 _H |

(h) ICERR — Instruction cache error register

This register stores cache error data of the instruction cache.

Once the ICHERR bit is set to 1, subsequent cache error data is not stored in this register until the ICHERR bit is cleared to 0. Also, the ICERR register is not updated while the cache is being accessed by the CILD instruction.

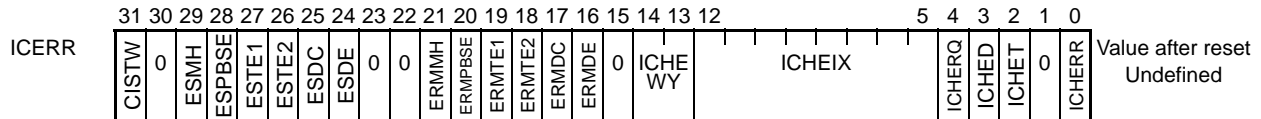


Table 3.57 ICERR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 | CISTW | This bit is set to indicate that the destination way specified for a CISTI instruction was in error. Although the entry information is overwritten so that writing is completed, the V bit will be cleared the next time the cache line is read (i.e. reading will be judged to have missed the cache). However, setting of this bit is not accompanied by an exception for the CPU. | R/W | 0 |
| 30 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 29 | ESMH | Error status: multi-hit | R/W | Undefined |
| 28 | ESPBSE | Error status: way error | R/W | Undefined |
| 27 | ESTE1 | Error status: 1-bit error in the tag RAM | R/W | Undefined |
| 26 | ESTE2 | Error status: 2-bit error in the tag RAM | R/W | Undefined |
| 25 | ESDC | Error status: 1-bit correction in the data RAM | R/W | Undefined |
| 24 | ESDE | Error status: 2-bit error in the data RAM | R/W | Undefined |
| 23, 22 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 21 | ERMMH | Error exception notification mask: multi-hit | R/W | 0 |
| 20 | ERMPBSE | Error exception notification mask: way error | R/W | 0 |
| 19 | ERMTE1 | Error exception notification mask: 1-bit error in the tag RAM | R/W | 0 |
| 18 | ERMTE2 | Error exception notification mask: 2-bit error in the tag RAM | R/W | 0 |
| 17 | ERMDC | Error exception notification mask: 1-bit correction in the data RAM | R/W | 0 |
| 16 | ERMDE | Error exception notification mask: 2-bit error in the data RAM | R/W | 0 |
| 15 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 14, 13 | ICHEWY | These bits retain a way number where a cache error occurs. | R/W | Undefined |
| 12 to 5 | ICHEIX | These bits retain a cache index where a cache error occurs. | R/W | Undefined |
| 4 | ICHERQ | Setting of this bit to 1 indicates that the CPU is being notified of a cache error exception. If cache error exceptions are masked, however, the CPU is not notified of an exception even when this bit is set to 1. | R/W | 0 |
| 3 | ICCHED | This bit indicates that an error occurs in the data RAM. | R/W | 0 |
| 2 | ICHET | This bit indicates that an error occurs in the tag RAM. | R/W | 0 |
| 1 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 0 | ICHERR | This bit is set to 1 when a cache error occurs. | R/W | 0 |

(7) Data Buffer Operation Function Registers

(a) Data buffer control function system registers

Data buffer control system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.58 Data Buffer Operation Function Registers

| Register No. (reg.ID, sel.ID) | Symbol | Function | Access Permission |
|----------------------------------|--------|------------------------------|----------------------|
| SR24,13 | CDBCR | Data buffer control register | SV |

(b) CDBCR — Data Buffer Control Register

This register controls the data buffer.

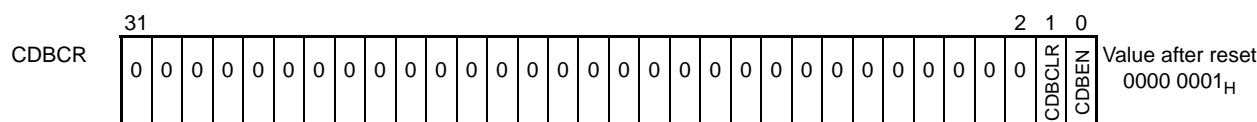


Table 3.59 CDBCR Register Contents

| Bit Position | Bit Name | Function | R/W | Value After Reset |
|--------------|----------|--|-----|-------------------|
| 31 to 2 | Reserved | (Reserved for future expansion. Be sure to clear to 0.) | R | 0 |
| 1 | CDBCLR | When this bit is set to 1, the data buffer is cleared at a time. This bit is always read as 0. | W | 0 |
| 0 | CDBEN | This bit disables or enables the data buffer. 0: Data buffer is disabled. 1: Data buffer is enabled. | R/W | 1 |

3.2.2 Instruction Cache and Data Buffer

3.2.2.1 Features

A 16-Kbyte and 4-way set-associative Instruction Cache is mounted between the CPU1 and the Code Flash. The Instruction Cache and the Code Flash are connected to each other via a 128-bit dedicated bus to minimize penalties caused by a cache miss-hit. Also a Data Buffer is mounted between the CPU1 and the Code Flash to achieve high-speed data access. The 32-MB area from 0000 0000_H to 01FF FFFF_H in the address space is intended for the Instruction Cache and Data Buffer.

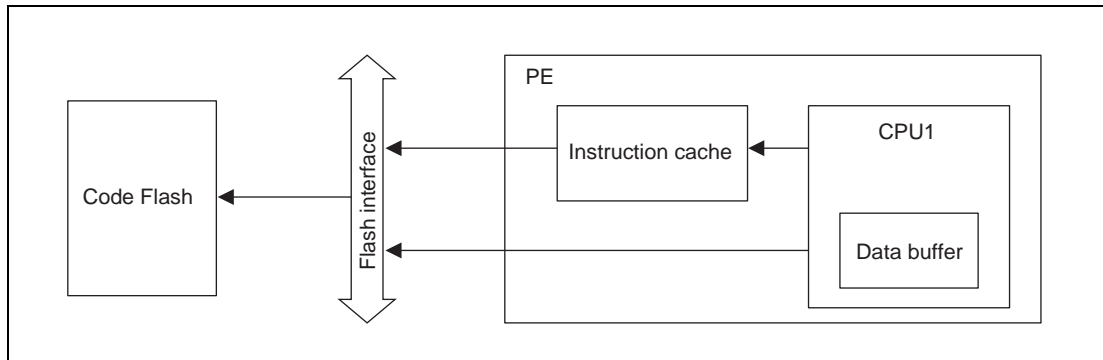


Figure 3.2 Instruction Cache and Data Buffer

3.2.2.2 Instruction Cache Function

The 16-Kbyte and 4-way set-associative cache includes four ways consisting of 256-entry blocks of four words per line, amounting to 16-Kbyte capacity in total. The ways are divided into two groups, way group 0 consisting of way0 and way1 and way group 1 consisting of way2 and way3. The way group can be selected and used by decoding of the address information of the access destination. If a cache error occurs, each line is refilled by a replace algorithm using LRU. CPU instruction fetches from the Code Flash are performed via the instruction cache.

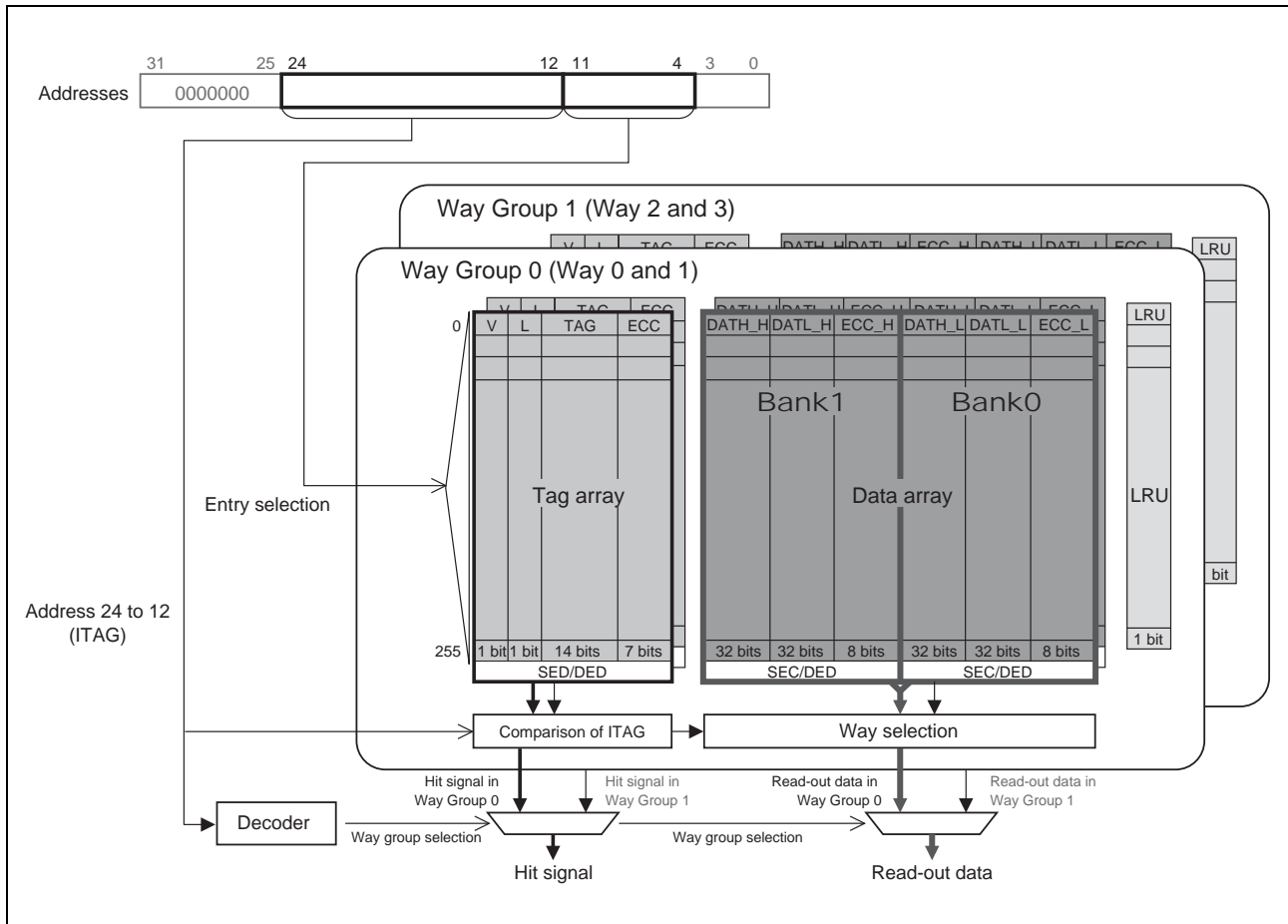


Figure 3.3 Instruction Cache Configuration

Tag Array

- V bit** This bit indicates whether valid data is stored in the cache line. Setting of this bit to 1 makes the cache line data valid. The V bit is initialized to 0 by reset.
- L bit** This bit indicates whether a cache line is locked or not. Setting of this bit to 1 locks the cache line and it can not be replaced with new data. The L bit is valid only when the V bit is 1, and it is not initialized by reset.
- TAG** Among 32 bits in the operable addresses of the data line to be cached, bits 24 to 12 are stored in this bit. The TAG bit is not initialized by reset.
- ECC** The ECC of the tag array is stored in this bit. The ECC bit is not initialized by reset.

Data Array

| | |
|---|---|
| DATH_H, DATH_L, DATL_H, DATL_L | The 128-bit cache line data is stored per 32 bits as follows: the bits [127:96], [95:64], [63:32], [31:0] in the DATH_H, DATL_H, DATH_L, and DATL_L, respectively. In the CIST or CILD operation in response to a cache instruction, the ICDATH register is used for DATH_H and DATH_L and ICATL is used for DATL_H and DATL_L. |
| ECC_H, ECC_L | The ECC of the data in bits [127:64] and [63:0] are stored in ECC_H and ECC_L, respectively. |

LRU

| | |
|-----|--|
| LRU | The LRU information in the same way group is stored in this data array. The LRU is initialized by reset. |
|-----|--|

CAUTION

When writing test data to the tag array of the instruction cache in the CIST instruction and then fetching the relevant line, write the tag information in units of way group. For example, when writing tag information to a line on the way 0 side of way group 0, also write the tag information of the same line on the way 1 side and then perform instruction fetch

- When writing data to way group 0 (way0, way1), write a value that makes the Exclusive OR of the ICTAGL.LPN bit 0.
- When writing data to way group 1 (way2, way3), write a value that makes the Exclusive OR of the ICTAGL.LPN bit 1.

When a value other than those above is written to the tag array and instruction fetch is performed, a way error occurs and the ICERR.ESPBSE bit is set to 1. When the same tag information is written to the same lines of two ways in a way group and instruction fetch is performed, a multi-hit error occurs and the ICERR.ESMH bit is set to 1.

3.2.2.3 Data Buffer Function

The four-line buffer with 128 bits per line is mounted as a data buffer. The data of 128 bits per line read from the Code Flash is stored in the data buffer. The data is read out from the data buffer after the next access to the same address, so the Code Flash is not accessed again.

3.2.3 Reliability Functions

3.2.3.1 PE Guard Function (PEG)

(1) Overview of the PEG Function

The PEG is a constituent of the slave guard system to prevent unauthorized access to the resources in the PE from the external master. This function protects access to the local RAM in the PE. In the initial state after a reset, all access by masters other than the PE itself is disabled. Setting the registers listed in **(3) List of PEG Protection Setting Registers**, enables access by masters other than the PE itself.

- Detecting PE guard violation

If the external master outside the PE makes an unauthorized access to the resource area in the PE for which PE guard is set, the access is detected as a PE guard violation. PE Guard informs it to ECM and reports the details about the access in registers.

- Blocking unauthorized accesses

When a PE guard violation is detected, unauthorized accesses to the internal resources of the PE are blocked to prevent the contents of PE resources from being modified illegally.

- Notifying occurrence of violation

An error response to an unauthorized access is sent to the request source of external master (except H-Bus master peripherals). When the DMAC or DTS makes an unauthorized access, meanwhile, a DMA transfer error is detected.

(2) Protection Made by SPID

- Setting PEG Protection

- Up to eight areas can be set depending on the Local RAM address of the own PE.
- The area range is specified by the base address and the mask bit (4 Kbytes to 4 Gbytes).
- “Read enable” and “write enable” can be set for each area.
- “Enable” or “disable” can be selected on each system protection identifier (SPID) for each area.

- Access permission by the system protection identifier (SPID) (see **Figure 3.4**)

1. When any of enabled area 0 to area 7 is to be accessed, go to step 2. Otherwise, return an error response.
2. Are all the conditions below for the relevant area met?
 - The system protection identifier (SPID) is enabled.
 - Required operations (read/write) are enabled.
 Otherwise, return an error response.

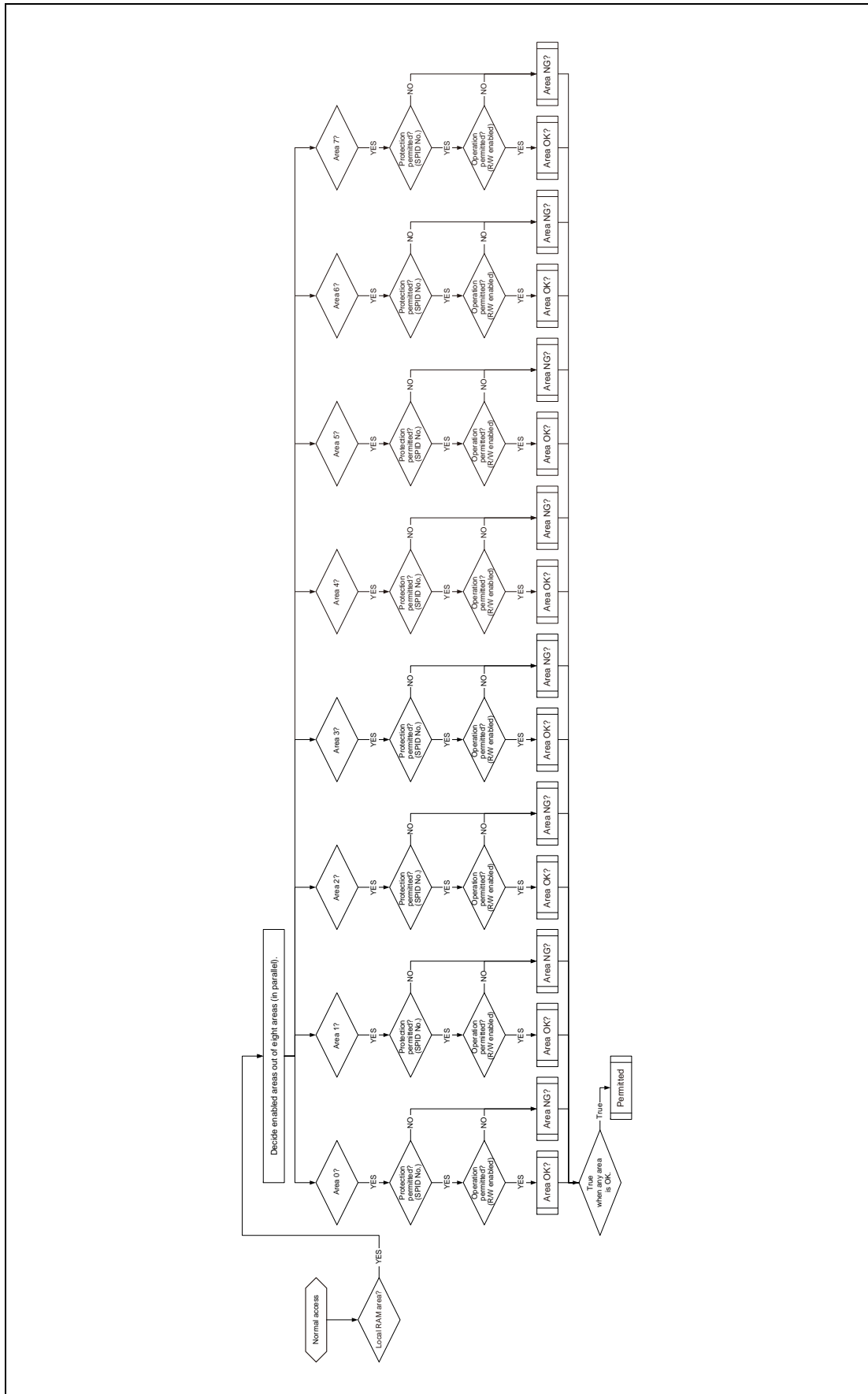


Figure 3.4 Access Permission by the System Protection Identifier (SPID)

(3) List of PEG Protection Setting Registers

Make necessary settings for the following registers to protect PE resources from unauthorized accesses by the external master.

- Accesses to the Local RAM in the PE are permitted as detection targets.
- For accesses to the register set, no access restriction is provided independently for the PEG function. Set access protections such as IPG at the user side as necessary.
- To set PEG protection, follow the procedure below. (n = 0 to 7)
 - Set the PE guard area n mask setting register as PEGGnMK.
 - Set the PE guard area n base setting register as PEGGnBA.

Table 3.60 PEG Protection Setting Registers (Base Address: FFFE E600_H)

| Address Offset | Size (Byte) | Register Name | Abbreviation | Right | R/W | Operable Bit | | | | Value After Reset |
|-------------------|-------------|---|--------------|-------|-----|--------------|---|----|----|------------------------|
| | | | | | | 1 | 8 | 16 | 32 | |
| +080 _H | 4 | PE guard area 0 mask setting register | PEGG0MK | — | R/W | — | √ | √ | √ | 003F F000 _H |
| +084 _H | 4 | PE guard area 0 base setting register | PEGG0BA | — | R/W | — | √ | √ | √ | FE80 0003 _H |
| +088 _H | 4 | PE guard area 0 enable setting register | PEGG0SP | — | R/W | — | √ | √ | √ | 0000 0003 _H |
| +090 _H | 4 | PE guard area 1 mask setting register | PEGG1MK | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +094 _H | 4 | PE guard area 1 base setting register | PEGG1BA | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +098 _H | 4 | PE guard area 1 enable setting register | PEGG1SP | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0A0 _H | 4 | PE guard area 2 mask setting register | PEGG2MK | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0A4 _H | 4 | PE guard area 2 base setting register | PEGG2BA | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0A8 _H | 4 | PE guard area 2 enable setting register | PEGG2SP | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0B0 _H | 4 | PE guard area 3 mask setting register | PEGG3MK | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0B4 _H | 4 | PE guard area 3 base setting register | PEGG3BA | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0B8 _H | 4 | PE guard area 3 enable setting register | PEGG3SP | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0C0 _H | 4 | PE guard area 4 mask setting register | PEGG4MK | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0C4 _H | 4 | PE guard area 4 base setting register | PEGG4BA | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0C8 _H | 4 | PE guard area 4 enable setting register | PEGG4SP | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0D0 _H | 4 | PE guard area 5 mask setting register | PEGG5MK | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0D4 _H | 4 | PE guard area 5 base setting register | PEGG5BA | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0D8 _H | 4 | PE guard area 5 enable setting register | PEGG5SP | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0E0 _H | 4 | PE guard area 6 mask setting register | PEGG6MK | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0E4 _H | 4 | PE guard area 6 base setting register | PEGG6BA | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0E8 _H | 4 | PE guard area 6 enable setting register | PEGG6SP | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0F0 _H | 4 | PE guard area 7 mask setting register | PEGG7MK | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0F4 _H | 4 | PE guard area 7 base setting register | PEGG7BA | — | R/W | — | √ | √ | √ | 0000 0000 _H |
| +0F8 _H | 4 | PE guard area 7 enable setting register | PEGG7SP | — | R/W | — | √ | √ | √ | 0000 0000 _H |

Table 3.61 PE guard Error Status Registers (Base Address of PE1: FFC4 A200_H)

| Address Offset | Register Name | Register Symbol | R/W | Value After Reset |
|------------------|--|------------------|-----|------------------------|
| +00 _H | PE guard Error Status Control Register | PGERRSTATCTL_PE1 | R/W | 0000 0000 _H |
| +04 _H | PE guard Error Status Register | PGERRSTAT_PE1 | R | 0000 0000 _H |
| +08 _H | PE guard Error Information Register | PGERRINFO_PE1 | R | 0000 0000 _H |

(4) Register Set

(a) PEGGnMK — PE Guard Area n Mask Setting Register (n = 0 to 7)

In combination with the PEGGnBA register, this register specifies a range or ranges within PE guard protection area n. Setting a GnMASK bit to 1 masks the corresponding address bit in the PEGGnBA register and places the corresponding area or area inside the range of PE guard area n. The minimum size of the PE guard protection area is 4KB.

Ex) With the settings of PEGGnBA[31:12]=FEBF6_H and PEGGnMK[31:12]=00008_H, the PE guard protection area n is specified within the ranges from FEBF 6000_H to FEBF 6FFF_H and FEBF E000_H to FEBF EFFF_H.

| | | | | | | | | | | | | | | | | |
|-------------------------|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GnMASK | | | | | | | | | | | | | | | |
| Value after reset *1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GnMASK | | | | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset *1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R |

Note 1. Refer to **Table 3.60, PEG Protection Setting Registers (Base Address: FFFE E600_H)**.

Table 3.62 PEGGnMK Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 12 | GnMASK | 0: Target address bits are compared when determining the PE guard area. 1: Target address bits are not compared when determining the PE guard area. |
| 11 to 0 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |

(b) PEGGnBA — PE Guard Area n Base Setting Register (n = 0 to 7)

In combination with the PEGGnMK register, this register specifies a range or ranges within PE guard protection area n. Setting the GnEN bit to 1 brings the address enable conditions specified by this register and the PEGGnMK register into effect.

| | | | | | | | | | | | | | | | | |
|---------------------------------|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|-----|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GnBASE | | | | | | | | | | | | | | | |
| Value after reset ^{*1} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GnBASE | | | | — | — | — | — | — | — | — | GnLOCK | — | GnWR | GnRD | GnEN |
| Value after reset ^{*1} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | W | R | R/W | R/W | R/W |

Note 1. Refer to **Table 3.60, PEG Protection Setting Registers (Base Address: FFFE E600_H)**.

Table 3.63 PEGGnBA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 12 | GnBASE | Base address that specifies PE guard protection area n |
| 11 to 5 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 4 | GnLOCK | Enables write protect to the registers (PEGGnBA, PEGGnMK, PEGGnSP) of PE guard control. This bit can be written once after reset. 0: Registers are not protected. 1: Registers are protected. |
| 3 | Reserved | This bit is always read as 0. The write value should always be 0. |
| 2 | GnWR | Enables write access to PE guard protection area n. 0: Write access is disabled. 1: Write access is enabled. |
| 1 | GnRD | Enables read access to PE guard protection area n. 0: Read access is disabled. 1: Read access is enabled. |
| 0 | GnEN | Enables the access enable condition settings for the PE guard protection area n. 0: Settings for access enable conditions are disabled 1: Settings for access enable conditions are enabled |

CAUTION

PEGGnBA.GnEN is cleared by writing to the PEGGnMK register.

(c) PEGGnSP — PE Guard Area n enable Setting Register (n = 0 to 7)

| | | | | | | | | | | | | | | | | |
|---------------------------------|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GnSPm | | | | | | | | | | | | | | | |
| Value after reset ^{*1} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GnSPm | | | | | | | | | | | | | | | |
| Value after reset ^{*1} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. Refer to **Table 3.60, PEG Protection Setting Registers (Base Address: FFFE E600_H)**.

Table 3.64 PEGGnSP Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | GnSPm | Set bit m to 1 to enable the access to the region n by the SPID m. (m = 0 to 31, shows bit position) 0: Access with SPID is not allowed. 1: Access with SPID is allowed. |

(d) PGERRSTATCTL_PE1 — PE Guard Error Status Control Register

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLRO | CLRE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 3.65 PGERRSTATCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When written, write value after reset. Read value is always value after reset. |
| 1 | CLRO | Clear the OVF bit of PGERRSTAT_PE1 by writing this bit to "1". 0: Clear is completed. 1: Clear is on execution. |
| 0 | CLRE | Clear the ERR bit of PGERRSTAT_PE1 by writing this bit to "1". 0: Clear is completed. 1: Clear is on execution. |

(e) PGERRSTAT_PE1 — PE Guard Error Status Register

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVF | ERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 3.66 PGERRSTAT_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When written, write value after reset. Read value is always value after reset. |
| 1 | OVF | 0: No or one access violation has occurred 1: More than one access violation has occurred |
| 0 | ERR | 0: No access violations have occurred 1: At least one access violation has occurred |

(f) PGERRINFO_PE1 — PE Guard Error Information Register

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|------|----|----|------|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | PEID | | | SPID | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 3.67 PGERRINFO_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 8 | Reserved | When written, write value after reset. Read value is always value after reset. |
| 7 to 5 | PEID | PEID of the bus master that initiated the access |
| 4 to 0 | SPID | SPID of the bus master that initiated the access |

3.2.3.2 PE's Internal Peripherals Protection Function (IPG)

(1) Overview of the IPG Function

The IPG is a system to prevent unauthorized accesses to peripherals from the CPU core equipped with the IPG. The IPG achieves the following functions. The IPG covers accesses to resources except the Code Flash and the local RAM.

step (1) Detecting violation of peripherals protection

If the CPU makes an unauthorized access to an area (peripherals) for which peripherals protection is set, the access is detected as "violation of peripherals protection."

step (2) Storing unauthorized-access information

When a violation of peripherals protection is detected, the unauthorized-access information is stored in the IPG's internal register.

step (3) Blocking unauthorized accesses

When a violation of peripherals protection is detected, unauthorized accesses to peripherals are blocked to prevent contents of peripherals from being modified illegally.

step (4) Notifying violation

When a violation of peripherals protection is detected, a system error exception (SYSERR exception) request for generating an exception is made to ask the CPU to stop the processing.

For system error exceptions (SYSERR exceptions), see **Section 3.2.3.3, System Error Notification Control Function (SEG)**.

step (5) Invalidating subsequent accesses

When a violation of peripherals protection is detected, subsequent accesses (regardless of authorized or unauthorized accesses) are blocked until instructions from the CPU are received.

NOTE

Even if a request for generating an exception is immediately sent to the CPU in step (4) above, a subsequent access issued (before receiving a request from the IPG) by the CPU that does not know an occurrence of violation may illegally modify contents of peripherals. (Accesses after a violation has occurred result in unauthorized accesses.)

(2) IPG Function

- This function invalidates accesses according to their attributes (including address, transfer type, and access right).
- After an access right violation is detected until the error flag (described later) is cleared by writing by the software, subsequent accesses are invalidated. However, invalidation is applied only to accesses from the CPU and is not applied to accesses from outside the CPU core. Invalidation is performed independently of addresses.
- When a request for accessing different peripherals simultaneously is made due to misalignment or double-word access, the access is executed when all such accesses are enabled.

(3) IPG Protection Setting Registers for Illegal Users

To protect peripherals from unauthorized accesses by programs in user mode, necessary settings are required for the registers listed below.

- Accesses in user mode are to be detected.
- This register set is intended for IPG settings related to user mode and reading the IPG settings in own machine.

Table 3.68 IPG registers (Base Address: FFFE E000_H)

| Address Offset | Size (Byte) | Register Name | Abbreviation | Right ^{*1} | R/W | Operable Bit | | | | Value After Reset |
|-------------------|-------------|--|--------------|---------------------|-----|--------------|---|----|----|----------------------|
| | | | | | | 1 | 8 | 16 | 32 | |
| +002 _H | 2 | Peripherals protection violation access information register | IPGECRUM | SV | R/W | — | — | √ | — | Undefined (retained) |
| +008 _H | 4 | Peripherals protection violation access address register | IPGADRUM | SV | R/W | — | — | — | √ | Undefined (retained) |
| +00D _H | 1 | Peripherals protection enable register | IPGENUM | SV | R/W | √ | √ | — | — | 00 _H |
| +020 _H | 1 | Peripherals protection setting register 0 | IPGPMTUM0 | SV | R/W | √ | √ | — | — | 00 _H |
| +021 _H | 1 | Peripherals protection setting register 1 | IPGPMTUM1 | SV | R/W | √ | √ | — | — | 00 _H |
| +022 _H | 1 | Peripherals protection setting register 2 | IPGPMTUM2 | SV | R/W | √ | √ | — | — | 00 _H |
| +023 _H | 1 | Peripherals protection setting register 3 | IPGPMTUM3 | SV | R/W | √ | √ | — | — | 00 _H |
| +024 _H | 1 | Peripherals protection setting register 4 | IPGPMTUM4 | SV | R/W | √ | √ | — | — | 00 _H |

Note 1. Registers for which "SV" is described are accessible by accesses with SV right (UM = 0).

(4) Register Set**(a) IPGECRUM — Peripherals Protection Violation Access Information Register**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | DS | | | EX | WR | RD | VD | |
| Value after reset | 0 | 0 | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

x: Undefined (retained)

Table 3.69 Register Contents of IPGECRUM

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 15, 14 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 13 to 8 | Reserved | The read value is undefined. The write value should always be 0. |
| 7 to 4 | DS | These bits store the data size of access that made a violation. 1000: Double-word (8 bytes) 0100: Word (4 bytes) 0010: Half-word (2 bytes) 0001: Byte Other than above: RFU |
| 3 | EX | This bit is set to 1 when a violation occurred in an instruction fetch read access. In other cases, this bit is cleared to 0. |
| 2 | WR | This bit is set to 1 when a violation occurred in a write access or bit operation or CAXI. In other cases, this bit is cleared to 0. |
| 1 | RD | This bit is set to 1 when a violation occurred in a read access or bit operation or CAXI. In other cases, this bit is cleared to 0. |
| 0 | VD | This bit is set to 1 when a violation of peripherals protection is detected by a program with the relevant right. If another violation of peripheral protection is detected, data of this IPGECRUM register and the IPGADRUM register is updated. |

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripherals protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(b) IPGADRUM — Peripherals Protection Violation Access Address Register

| | | | | | | | | | | | | | | | | |
|-------------------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | EADR | | | | | | | | | | | | | | | |
| Value after reset | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EADR | | | | | | | | | | | | | | | |
| Value after reset | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

x: Undefined (retained)

Table 3.70 Register Contents of IPGADRUM

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | EADR | These bits store the address of the access in which a violation occurred. |

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripherals protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(c) IPGENUM — Peripherals Protection Enable Register

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | IRE | E |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 3.71 Register Contents of IPGENUM

| Bit Position | Bit Name | Function |
|---|----------|--|
| 7 to 2 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 1 | IRE | This bit sets whether to store the access information in the peripherals protection violation access address register and the peripherals protection violation access information register when a violation of peripherals protection occurred in an instruction fetch access. 0: Instruction fetch access information is not stored. (Value after reset) 1: Instruction fetch access information is stored. |
| CAUTION | | |
| If you do not want to detect speculative instruction fetches (no instruction is executed in some cases), clear this bit to 0. | | |
| 0 | E | This bit enables or disables the peripherals protection function against accesses by the relevant access right. 0: The peripherals protection function is disabled. (Value after reset) 1: The peripherals protection function is enabled. |

(d) IPGPMTUM0 — Peripherals Protection Setting Register 0

| | | | | | | | | |
|-------------------|---|-----|-----|-----|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | X1 | W1 | R1 | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R | R | R | R |

Table 3.72 Register Contents of IPGPMTUM0

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 6 | X1 | This bit sets whether to enable instruction fetch read access to Peripheral IP groups. 0: Instruction fetch read access to Peripheral IP groups is treated as violation. (Value after reset) 1: Instruction fetch read access to Peripheral IP groups is not restricted. |
| 5 | W1 | This bit sets whether to enable write access to Peripheral IP groups. 0: Write access to Peripheral IP groups is treated as violation. (Value after reset) 1: Write access to Peripheral IP groups is not restricted. |
| 4 | R1 | This bit sets whether to enable read access to Peripheral IP groups. 0: Read access to Peripheral IP groups is treated as violation. (Value after reset) 1: Read access to Peripheral IP groups is not restricted. |
| 3 to 0 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |

(e) IPGPMTUM1 — Peripherals Protection Setting Register 1

| | | | | | | | | |
|-------------------|---|-----|---|---|---|-----|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | X1 | — | — | — | X0 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R | R | R | R/W | R | R |

Table 3.73 Register Contents of IPGPMTUM1

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 6 | X1 | This bit sets whether to enable instruction fetch read access to GRAM Bank B. 0: Instruction fetch read access to GRAM Bank B is treated as violation. (Value after reset) 1: Instruction fetch read access to GRAM Bank B is not restricted. |
| 5 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | X0 | This bit sets whether to enable instruction fetch read access to GRAM Bank A. 0: Instruction fetch read access to GRAM Bank A is treated as violation. (Value after reset) 1: Instruction fetch read access to GRAM Bank A is not restricted. |
| 1, 0 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |

(f) IPGPMTUM2 — Peripherals Protection Setting Register 2

| | | | | | | | | |
|-------------------|---|---|-----|-----|---|---|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | W1 | R1 | — | — | W0 | R0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R/W | R/W |

Table 3.74 Register Contents of IPGPMTUM2

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7, 6 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 5 | W1 | This bit sets whether to enable write access to COMPTEST. 0: Write access to COMPTEST is treated as violation. (Value after reset) 1: Write access to COMPTEST is not restricted. |
| 4 | R1 | This bit sets whether to enable read access to COMPTEST. 0: Read access to COMPTEST is treated as violation. (Value after reset) 1: Read access to COMPTEST is not restricted. |
| 3, 2 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 1 | W0 | This bit sets whether to enable write access to INTC1. 0: Write access to INTC1 is treated as violation. (Value after reset) 1: Write access to INTC1 is not restricted. |
| 0 | R0 | This bit sets whether to enable read access to INTC1. 0: Read access to INTC1 is treated as violation. (Value after reset) 1: Read access to INTC1 is not restricted. |

(g) IPGPMTUM3 — Peripherals Protection Setting Register 3

| | | | | | | | | |
|-------------------|---|---|-----|-----|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | W1 | R1 | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R | R |

Table 3.75 Register Contents of IPGPMTUM3

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7, 6 | Reserved | When read, the value after reset always is read. When writing, always write the value after reset. |
| 5 | W1 | This bit sets whether to enable write access to SYSERRGen. 0: Write access to SYSERRGen is treated as violation. (Value after reset) 1: Write access to SYSERRGen is not restricted. |
| 4 | R1 | This bit sets whether to enable read access to SYSERRGen. 0: Read access to SYSERRGen is treated as violation. (Value after reset) 1: Read access to SYSERRGen is not restricted |
| 3 to 0 | Reserved | When read, the value after reset always is read. When writing, always write the value after reset. |

(h) IPGPMTUM4 — Peripherals Protection Setting Register 4

| | | | | | | | | |
|-------------------|---|-----|-----|-----|---|---|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | X1 | W1 | R1 | — | — | W0 | R0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R | R | R/W | R/W |

Table 3.76 Register Contents of IPGPMTUM4

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 6 | X1 | This bit sets whether to enable instruction fetch read access to peripherals to be connected to the H-Bus. 0: Instruction fetch read access to peripherals to be connected to the H-Bus is treated as violation. (Value after reset) 1: Instruction fetch read access to peripherals to be connected to the H-Bus is not restricted. |
| 5 | W1 | This bit sets whether to enable write access to peripherals to be connected to the H-Bus. 0: Write access to peripherals to be connected to the H-Bus is treated as violation. (Value after reset) 1: Write access to peripherals to be connected to the H-Bus is not restricted. |
| 4 | R1 | This bit sets whether to enable read access to peripherals to be connected to the H-Bus. 0: Read access to peripherals to be connected to the H-Bus is treated as violation. (Value after reset) 1: Read access to peripherals to be connected to the H-Bus is not restricted. |
| 3, 2 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 1 | W0 | This bit sets whether to enable write access to PEG. 0: Write access to PEG is treated as violation. (Value after reset) 1: Write access to PEG is not restricted. |
| 0 | R0 | This bit sets whether to enable read access to PEG. 0: Read access to PEG is treated as violation. (Value after reset) 1: Read access to PEG is not restricted |

3.2.3.3 System Error Notification Control Function (SEG)

Errors due to an instruction fetch or data access can be the sources of system error exceptions. A SYSERR exception is an FE level exception from which return or recovery is not possible.

SEG (SysErrGen) controls the notification and record of the error by the data access. Instruction cache errors occurred on the RAM are notified to the SEG. For details, see section (a) **SEGCONT — Error Notification Control Register** and (c) **Additional information on SYSERR exception**.

Multiple error occurrence inputs are categorized according to error factors, and are processed sequentially from the highest-priority error factor, generating an FE-level asynchronous exception (SYSERR).

The bit position of the SEGFLAG register becomes the priority of error factors. Error factors of lower bits take precedence over error factors of upper bits.

Error information is recorded once regardless of error frequency.

The error with the highest priority of error factor (in case errors occurred simultaneously) is valid. Recorded error information is not overwritten by subsequent errors.

(1) List of SEG Function Control Registers

Table 3.77 SEG Register (Base Address: FFFE E980_H)

| Address Offset | Size (Byte) | Register Name | Abbreviation | Right | R/W | Operable Bit | | | | Value After Reset |
|------------------|-------------|---|--------------|-------|--------|--------------|---|----|----|----------------------|
| | | | | | | 1 | 8 | 16 | 32 | |
| +00 _H | 2 | Error notification control register | SEGCONT | — | R/W *1 | — | — | √ | — | 0000 _H |
| +02 _H | 2 | Error occurrence retention register | SEGFLAG | — | R/W *1 | — | — | √ | — | 0000 _H |
| +08 _H | 4 | Error factor retention register (address) | SEGADDR | — | R/W *1 | — | — | √ | √ | Undefined (retained) |

Note 1. Write accesses from user mode are ignored.

NOTE

- If an access is made with an address offset or operable bits other than those specified above, an error response is returned.
- Write access is only possible with the SV privilege (UM = 0). Attempting to write, if these conditions do not hold, leads to an error response being returned.
- No restriction is provided for read accesses.
 - Read accesses to ranges permitted by other protection systems are enabled at any time.

(2) Register Set**(a) SEGCONT — Error Notification Control Register**

- This register is used to enable (= 1) or disable (= 0) notification of SYSERR request in response to error flags that store error occurrence status according to factors.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|------|------|---|------|---|------|---|------|---|-----|
| | — | — | — | — | — | — | VPGE | VCRE | — | TCME | — | VCIE | — | ICCE | — | NEE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R/W | R | R/W | R | R/W | R | R/W |

Table 3.78 SEGCONT Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 15 to 10 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 9 | VPGE | This bit notifies an error in writing access to P-Bus. The error includes the followings: <ul style="list-style-type: none"> P-Bus guard error Address EDC error Data ECC error Access to unimplemented area in the P-Bus |
| 8 | VCRE | This bit notifies the IPG violation access detection and subsequent access blocking (including instruction fetch). ^{*1} |
| 7 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 6 | TCME | This bit notifies an error in accessing data to local RAM. The error includes the following cases: <ul style="list-style-type: none"> ECC error Access to the RAM-unimplemented area in the local RAM |
| 5 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |

Table 3.78 SEGCONT Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 4 | VCIE | <p>This bit notifies an error from the GRAM, Code Flash, H-bus or P-bus. The error includes the followings:</p> <ul style="list-style-type: none"> • in reading access to P-Bus <ul style="list-style-type: none"> – P-Bus guard error – Address EDC error – Data ECC error – access to unimplemented area in the P-Bus • access to unimplemented area in the CPU-specific functions • in access to H-Bus <ul style="list-style-type: none"> – H-Bus guard error – access to unimplemented area in the H-Bus area • in access to Code Flash <ul style="list-style-type: none"> – Address Parity error – Data ECC error • in access to system peripheral <ul style="list-style-type: none"> – FFFF 0000_H to FFFF 4FFF_H – FFFE 0000_H to FFFE BFFF_H – FB00 0000_H to FE9F FFFF_H – F300 0000_H to F8FF FFFF_H • in access to Global RAM <ul style="list-style-type: none"> – GRAM guard error – Address ECC error – Data ECC error • This bit notifies of an IPG violation access detection and subsequent access blocking.*¹ • This bit notifies of an access privilege violation. <ul style="list-style-type: none"> – Read or write access to the IPG Protection Setting Registers by user mode (PSW.UM = 1). – Write access to the SEG Function Control Registers by user mode (PSW.UM = 1). |
| 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | ICCE | <p>Instruction cache error notification enable</p> <p>The error occurred in the instruction cache is handled when the instruction cache system register, ICCTRL.ICHEMK, is set to 0 (whose value after reset is 1):</p> <p>For instruction cache errors, see (h) ICERR — Instruction Cache Error Register in Section 3.2.1.2, Register Set</p> |
| 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | NEE | <p>This bit notifies the correctable ECC error. Code Flash, Global RAM, Local RAM.</p> <p>To enable this function, it is necessary to set VCIE or TCME bit to 1 at the same time.</p> |

Note 1. Please refer the IPGADRUM register of **Section 3.2.3.2, PE's Internal Peripherals Protection Function (IPG)** for an error factor addresses.

(b) SEGFLAG — Error Occurrence Retention Register

- The register SEGFLAG indicates from which slaves error responses have been received so far. SEGFLAG is not cleared automatically. Clearing can be performed by writing a zero into the register.
- Writing to the register enables both setting and clearing.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|------|------|---|------|---|------|---|------|---|---|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | VPGF | VCRF | — | TCMF | — | VCIF | — | ICCF | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R/W | R | R/W | R | R/W | R | R |

Table 3.79 SEGFLAG Register Contents

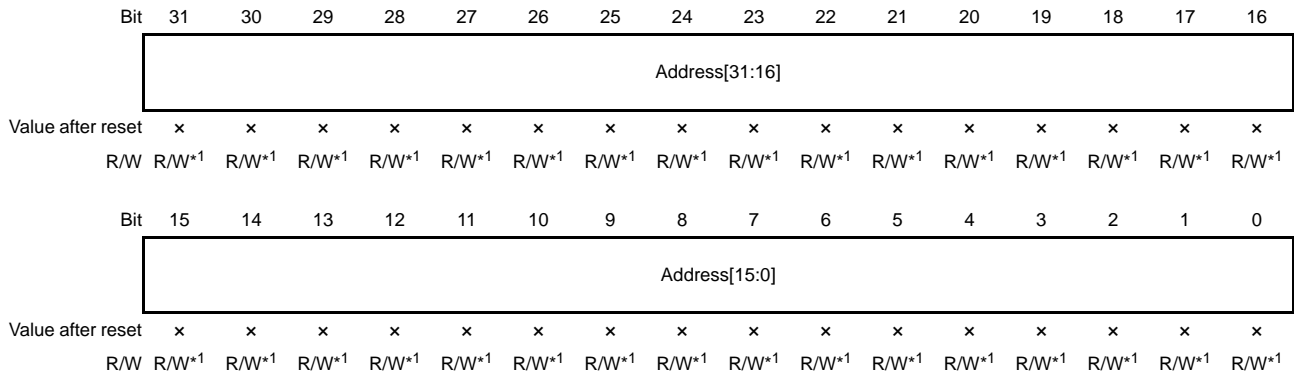
| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 15 to 10 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 9 | VPGF | Flag corresponding to bit 9 of the SEGCONT register |
| 8 | VCRF | Flag corresponding to bit 8 of the SEGCONT register |
| 7 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 6 | TCMF | Flag corresponding to bit 6 of the SEGCONT register |
| 5 | Reserved | When read, the undefined value is returned. When writing, write the value after reset. |
| 4 | VCIF | Flag corresponding to bit 4 of the SEGCONT register |
| 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | ICCF | Flag corresponding to bit 2 of the SEGCONT register |
| 1, 0 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |

Table 3.80 Relationship Between System Error Exception Cause Code and SEGFLAG

| FEIC | Error Cause |
|-----------------|--|
| 10 _H | Reserved |
| 11 _H | Instruction fetch from Code Flash |
| 12 _H | SEGFLAG.ICCF |
| 13 _H | Instruction fetch from other than Code Flash |
| 14 _H | SEGFLAG.VCIF |
| 15 _H | Reserved |
| 16 _H | SEGFLAG.TCMF |
| 17 _H | Reserved |
| 18 _H | SEGFLAG.VCRF |
| 19 _H | SEGFLAG.VPGF |

(c) SEGADDR — SEG Error Information Register (Address)

This register records information of an error source that notifies a SYSERR request (only one history is recorded). It records the addresses where the error factors for the VCIF and TCMF bits of the SEGFLAG register were found. The error factors except VCIF and TCMF bits will be recorded as 0000 0000_H. This register cannot be modified when the error occurrence flag to enable notification is set.



x: Undefined (retained)

Note 1. This register cannot be modified when the error occurrence flag to enable notification is set.

Table 3.81 SEGADDR Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31 to 0 | Address[31:0] | These bits hold the address at which SYSERR was generated. (If the error was caused by an access to the local RAM area, the lower 19 bits of the address are held, and the upper 13 bits are cleared to 0.) |

(3) SEG Function**(a) SEG function: Notifying a SYSERR request due to an error flag**

- Setting an error flag takes precedence over clearing the same flag.
 - Simultaneous clearing operation is ignored.
- Priority of error factors
 - The bit position of the notification-enabled SEGFLAG register becomes the priority of error factors. Error factors of lower bits take precedence over error factors of upper bits. Notification is made from the highest-priority error factor.
 - The bit position of error factors is notified as a “SYSERR factor code.”
- Conditions for starting SYSERR request notification
 - Even if a SEGFLAG register bit is set to 1, notification is not made.
 - Notification is made immediately after a SEGCONT register bit is set to 1.
 - After clearing SEGFLAG register bits of error notification subject, notification is made depending on priority of SEGFLAG register bits except for cleared bits (re-arbitration).
- Finishing notification at a SYSERR request response
 - Even after notification is finished, the flag is not cleared automatically.
 - Notification is not made until re-arbitration is performed by setting or clearing the flag.
 - If an error flag that is prioritized higher than the error factor is set prior to a request response, the notification information may be replaced with an upper SYSERR factor code.

(b) SEG function: Recording error factor information

- When notification-enabled error occurrence is input, the error address is retained in the above register.
 - No information is retained by setting or clearing an error flag described in “**(3) (a) SEG function: Notifying a SYSERR request due to an error flag**” above.
 - When multiple error occurrence inputs are present simultaneously, information other than the prioritized error factor is not retained.
- While the notification-enabled error flag described in “**(3) (a) SEG function: Notifying a SYSERR request due to an error flag**” above is set to 1, overwrite to the above register is inhibited
 - If error occurrence input continues, information of subsequent error factors is not retained.
 - To cancel the register overwrite prohibition, clear either SEGCONT or SEGFLAG register (or both of them).

(c) Additional information on SYSERR exception

- Even if an SYSERR exception occurs, the value of the PSW.EBV bit is kept and the base address of an exception handler does not change.
- Error detection in instruction cache

Even if an error is detected in instruction cache, resumable SYSERR exception by instruction fetch factor does not occur. Instruction cache automatically invalidates the entry which includes an error and the CPU continues the instruction execution by re-fetching from the Code Flash. When the ICCTRL.ICHEMK bit in the system register is set to 0, an error occurred in instruction cache is notified to the SEG.

3.2.3.4 Checker Core

The CPU1 (PE1) has the Checker Core for safety assurance, resulting in a highly reliable system. Monitoring the outputs from the CPU1 (PE1) and the Checker Core with the comparator all the time enables immediate detection of the CPU1 (PE1) abnormal operations. Duplication by the checker core covers the CPU core, FPU, MPU, PEG, IPG, SEG, and INTC1. The CPU1 can also conduct a fault diagnosis test of its own comparator through a pseudo-error generated by the COMPTEST module. Please refer to **Section 31, Functional Safety** for detail.

CAUTION

Reading of any register with an undefined value after a reset in a PE or writing to memory or a register outside the PE may cause a lock step comparison error. The values of some program registers and system registers are undefined after a reset, so pay attention to this when saving register values on the stack in RAM.

3.3 Inter CPU Functions Overview

3.3.1 Processor Element Identifier

The PEID, each processor element ID number, can be read from the PEID field in the HTCFCG0 register. Which CPU core performs a specific program can be understood by referring to the PEID. The following shows the PEID of this product.

| CPU core | PEID |
|------------|------------------|
| CPU1 (PE1) | 001 _B |

3.3.2 Exclusive Function

The Local RAM and Global RAM are available as a resource for exclusive control. Atomic operation instructions LDL/STC, CAXI, SET1, CLR1, and NOT1 can be performed for the Local RAM and the Global RAM. Such instructions can not be performed to H-Bus area.

3.3.3 Write-Through Buffer for Global RAM

It is accessible by CPU core through a 64-bit bus. CPU1 (PE1) has a Write-Through Buffer (WT-Buffer or WT-Buf for short). The WT-Buf consists of 2 banks (one for each GRAM bank) and each bank consists of 8 entries x 64bits. The WT-Buf behavior is summarized below according to access type from PE:

Read hit: data are returned from the WT-Buf.

Read miss: data are read from GRAM and they are registered in WT-Buf

Write hit: Both WT-Buf and GRAM are updated

Write miss (no RmW): Both WT-Buf and GRAM are updated

Write miss (with RmW): Read from GRAM first and write the parts to both WT-Buf and GRAM

The data coherency between GRAM and WT-Buffer is always kept because the same data is always written to GRAM and WT-Buffer at the same time. The data coherency between PE and DMA is also kept, as DMA writes to GRAM as well as WT-Buffer at the same time and DMA always reads from GRAM not from WT-Buffer.

The WT-Buffer equips invalidation timer to invalidate the entries automatically. Invalidation timer cycle count can be chosen from 16-cycles/64-cycles/256-cycles. Software invalidation is also possible and one register is provided for this purpose.

The WT-Buffer configuration itself can also be chosen either On or Off. After reset, WT-Buffer is set as On.

(1) List of Write-Through Buffer Control Registers**Table 3.82 Write-Through Buffer Control Register (Base Address: FFFF 7A00_H)**

| Address | Register Name | Function | Access Width | Value after reset |
|--------------------------------------|-----------------|--------------------------------------|--------------|------------------------|
| <GRAMC_QOSREG_base> + 4 _H | GRAMCWTBCONFIG0 | GRAMC WTBuf Configuration Register 0 | 32bit | 0000 0001 _H |
| <GRAMC_QOSREG_base> + 8 _H | GRAMCWTBCONFIG1 | GRAMC WTBuf Configuration Register 1 | 32bit | 0000 0007 _H |
| <GRAMC_QOSREG_base> + C _H | GRAMCWTBCONFIG2 | GRAMC WTBuf Configuration Register 2 | 32bit | 0000 0000 _H |

(2) Register Sets**(a) GRAMCWTBCONFIG0 — GRAMC WTBuf Configuration Register 0**

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WTbuf Mode |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 3.83 GRAMCWTBCONFIG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | WTBufMode | This bit sets whether to enable WT Buf 0: disable WT Buf 1: enable WT Buf |

CAUTION

The value of GRAMCWTBCONFIG0.WTBufMode bit may be changed only when all bus masters are not access to Global RAM.

(b) GRAMCWTBCONFIG1 — GRAMC WTBuf Configuration Register 1

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|------------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | WTBufTmrCmp | WTBufTmrEn | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 3.84 GRAMCWTBCONFIG1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 to 1 | WTBufTmrCmp | These bits set WT Buf invalidate timer count number. 00: 16 cycles 01: 32 cycles 10: 64 cycles 11: 256 cycles |
| 0 | WTBufTmrEn | This bit sets whether to enable WT Buf invalidation timer. 0: disable WT Buf invalidate timer count down 1: enable WT Buf invalidate timer count down All WT Buf entries are flushed when this bit is set from 1 to 0. |

(c) GRAMCWTBCONFIG2 — GRAMC WTBuf Configuration Register 2

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WTBufTmcClr |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R ¹ |

Note 1. The function of this bit is used by read access.

Table 3.85 GRAMCWTBCONFIG2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | WTBufTmcClr | When this register is read, All WT Buf entries are flushed. The read value of this bit is always 0. |

3.4 Usage Notes

3.4.1 Synchronization of Store Instruction Completion and Subsequent Instruction Generation

When a control register is updated by a store instruction, there is a time lag after the CPU executes the store instruction and before the control register is actually updated. Therefore, if the updated content of the control register is to be used by the instruction following the store instruction, the appropriate synchronization is required. How to perform synchronization processing is shown below. For the procedures to synchronize updating system registers by LDSR instruction and the subsequent instruction execution, see APPENDIX A. HAZARD RESOLUTION PROCEDURE FOR SYSTEM REGISTERS in the RH850G3M User's Manual: Software.

3.4.1.1 When updated results in the control registers are reflected in the implementation of a subsequent instruction:

[Example 1]

This includes the following case: an interrupt is enabled by implementation of an EI instruction after an interrupt request is cleared by access from the control register in the INTC2 and the peripheral circuits. Proceed as follows in this case.

1. Execute the store instruction to update a control register (ST.W, etc.).
2. Perform a dummy read of the above control register (LD.W etc.).
3. Execute SYNC.P.
4. Execute the subsequent instruction (EI).

[Example 2]

Implement the same processing even when the access required after waiting to secure the updating of a given control register (register A) is to another control register (register B). This includes the following cases: the interlinked operation of different peripheral modules and when releasing the interrupt mask in INTC after making peripheral module settings. However, this processing is unnecessary if control registers A and B are in the same group of peripheral IP modules or control registers A and B are both in the CPU's own peripheral modules.

1. Execute the store instruction to update control register A (ST.W, etc.).
2. Perform a dummy read of the above control register (LD.W, etc.).
3. Execute SYNC.P.
4. Execute the store instruction to access control register B (ST.W, LD.W, etc.).

The same processing is also required when access to control registers and memory within the scope of protection starts after waiting for the completion of settings for safety functions such as memory protection, ECC checking, and so on.

For details of the peripheral groups, see **Section 1.4, Block Configuration**.

For the procedures to synchronize updating system registers by LDSR instruction and the subsequent instruction execution, see *APPENDIX A. Hazard Resolution Procedure for System Registers* in the *RH850G3M User's Manual: Software*.

3.4.1.2 When the updated results of the control registers or memory to be used in the instruction fetch of the subsequent instruction:

- (a) In case of writing the instructions to the RAM before jumping to the RAM to execute instructions from the RAM, take the following procedure.
 1. Execute the store instruction to update a memory (ST.W, etc.).
 2. Perform a dummy read of the above-mentioned memory (LD.W, etc.).
 3. Execute SYNCP.
 4. Execute SYNCL.
 5. Execute the subsequent instruction (branch instruction, etc.).
- (b) In case of updating control registers for memory protection and ECC functions before jumping to the memory to be controlled by the registers, take the following procedure.
 1. Execute the store instruction to update a control register (ST.W, etc.).
 2. Perform a dummy read of the above-mentioned control register (LD.W, etc.).
 3. Execute SYNCP.
 4. Execute SYNCL.
 5. Execute the subsequent instruction (branch instruction, etc.).

3.4.1.3 When switching the code flash memory area:

In this case, see *Section 9, Usage Notes, (7) Update of BFASCLR register* in the *RH850/P1M-E Flash Memory User's Manual: Hardware Interface*.

3.4.2 Accesses to Registers by Bit-Manipulation Instructions

Writing by using bit-manipulation instructions consists of atomic read-modify-write processing in 8-bit units. Thus, access by a bit-manipulation instruction is only possible for registers for which reading and writing in 8-bit units is possible. If a register includes multiple flag bits, the read-modify-write operation may lead to the clearing of flags that were not actually targets for clearing.

Write access to the FlexRay registers by using bit-manipulation instructions is not atomic. Access by other masters may interrupt the read-modify-write processing of these instructions.

3.4.3 Ensuring Coherency after Code Flash Programming

The CPU has an efficient instruction cache and data buffer for the code flash area.

Therefore, after using self-programming to program the code flash memory, clear the instruction cache and data buffer to ensure coherency. The instruction cache and data buffer can be cleared by using the ICCTRL register and the CDBCR register, respectively.

3.4.4 Overwriting Context when Acknowledging Multiple Exceptions

Acceptance of an exception depends on the type of exception source, regardless of the states of the ID and NP bits in the PSW register. When multiple exceptions are generated, the contents of the system register which hold the context information are overwritten. For the conditions for acceptance and whether correct return or recovery is possible for each exception source, see the List of Exception Sources in the *RH850G3M User's Manual: Software*.

3.4.5 Usage Notes on Prefetching

CPU executes speculative instruction fetching from locations later than the current value of the program counter to maintain the throughput of instruction fetches. Reading from memory due to such prefetching may proceed even from locations to which instruction codes have not been assigned (note 1 in **Figure 3.5**). Please note the following. The CPU does not execute values read in such cases.

These notes apply to instruction fetching from memory in general.

- Occurrence of ECC errors due to values in memory being undefined

This prefetching may lead to an ECC error in case of reading from the code flash memory after it has been erased or from the local RAM before initialization. When instruction codes are assigned to memory, initialize said area with values as desired (note 1 in **Figure 3.5**).

- Detection of illegal access by the IPG

The IPG may detect such prefetching as illegal access. To prevent prefetching being detected as an illegal access, do not allow any region of overlap area with said areas (note 1 in **Figure 3.5**) and areas to which access is prohibited by the IPG. Reading from an area protected by the MPU does not cause a memory protection exception.

- Access to Access Prohibited Area

Assign instruction codes to memory without allowing any overlap between said area (note 1 in **Figure 3.5**) and an access-prohibited area.

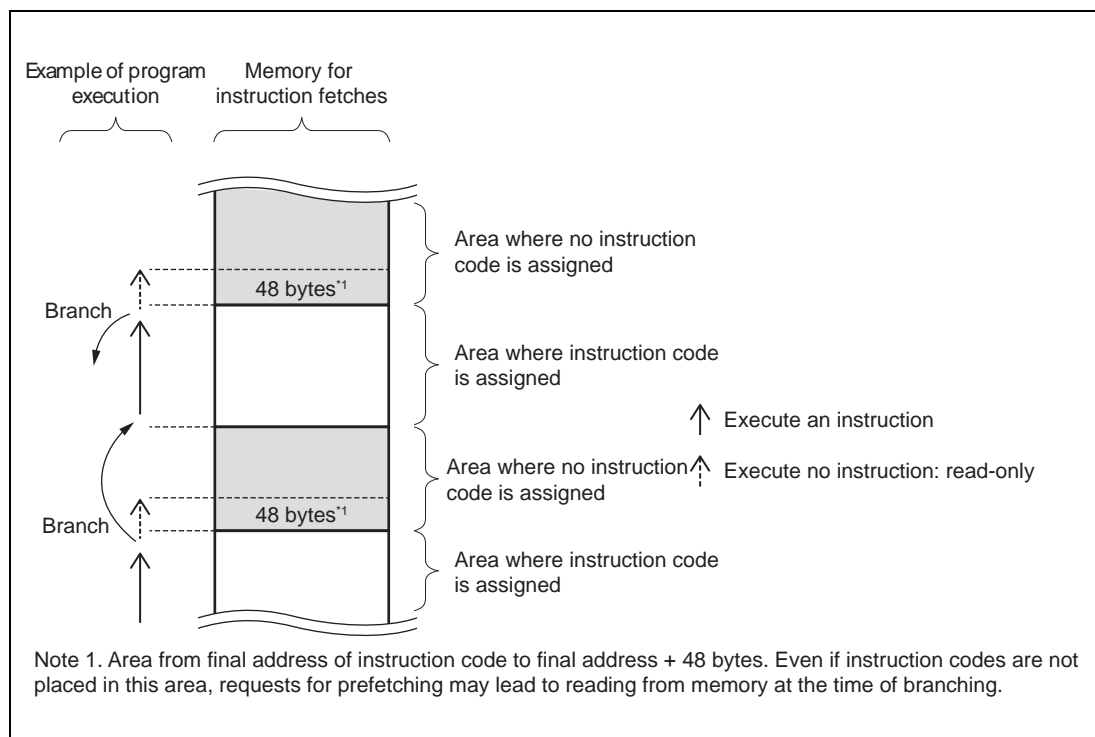


Figure 3.5 Area that Requires Attention Regarding Prefetching

3.4.6 Usage Note when Exception is Acknowledged

SYNCP instruction must be inserted in front of exception handler, depending on the exception sources.

For the details, see the *RH850G3M User's Manual: Software*.

Section 4 Address Space

Table 4.1 shows the address space of the RH850/P1M-E.

CAUTION

When making an access to the on-chip I/O register space, access the addresses shown in this manual. Do not access an address that is reserved or not specified in this manual. Otherwise, operation is not guaranteed.

4.1 Address Space

Table 4.1 Address Space

| Address | Address Space Type | Size |
|---|---|---------------------------------|
| 0000 0000 _H to * ¹ | Code Flash (user area) | 1 MB/2 MB |
| * ¹ to 00FF FFFF _H | Reserved | |
| 0100 0000 _H to 0100 7FFF _H | Code Flash (extended user area) | 32 KB |
| 0100 8000 _H to 0100 9FFF _H | Reserved | |
| 0100 A000 _H to 0100 BFFF _H | ECC test area | 8 KB |
| 0100 C000 _H to 0FFF FFFF _H | Reserved | |
| 1000 0000 _H to 1FFF FFFF _H | On-chip I/O register (H-Bus Area) | 256 MB |
| 2000 0000 _H to FEBD FFFF _H | Reserved | |
| FEBE 0000 _H to FEBF FFFF _H | Local RAM (PE1 area) | 128 KB |
| FEC0 0000 _H to FEDD FFFF _H | Reserved | |
| FEDE 0000 _H to FEDF FFFF _H | Local RAM (self) | 128 KB |
| FEE0 0000 _H to FEEF 7FFF _H | Reserved | |
| FEEF 8000 _H to FEEF FFFF _H | Global RAM (Bank A) | 32 KB |
| FEF0 0000 _H to FEF0 7FFF _H | Global RAM (Bank B) | 32 KB |
| FEF0 8000 _H to FEFF FFFF _H | Reserved | |
| FF00 0000 _H to FFFD FFFF _H (FF20 0000 _H to * ³) | On-chip I/O register (P-Bus Area) (Data flash) | 16 MB – 128 KB (32 KB/64 KB) |
| FFFE 0000 _H to FFFE DFFF _H | Reserved | |
| FFFE E000 _H to FFFE FFFF _H | On-chip I/O register (LPB Area self* ²) | 8 KB |
| FFFF 0000 _H to FFFF 4FFF _H | Reserved | |
| FFFF 5000 _H to FFFF FFFF _H | On-chip I/O register (P-Bus Area) | 44 KB |

Note 1. 1-MB device: 000F FFFF_H, 2-MB device: 001F FFFF_H

Note 2. The "self" on-chip I/O registers are in an area to which CPU-specific functions (SEG, PEG, IPG, INTC1) are allocated. This area can only be accessed by CPU1 (PE1).

Note 3. 1-MB device: FF20_7FFF_H, 2-MB device: FF20_FFFF_H

4.2 Address Space Viewed from Each Bus Master

Figure 4.1 shows address spaces viewed from each bus master.

4.2.1 Space in which instructions can be fetched

1. Instructions of PE1 can be fetched from the code flash, the local RAM (self) and the global RAM.
2. The reset vector (RBASE initial value) of PE1:
 - When the startup mat is the user mat, its head address is 0000 0000_H.

4.2.2 Data space accessible by PE1

All spaces are accessible.

4.2.3 Data space accessible by DMA (DMAC, DTS)

See Figure 4.1 for the accessible spaces from the DMA.

4.2.4 Data space accessible by H-Bus master

See **Figure 4.1** for the accessible spaces from the H-Bus.

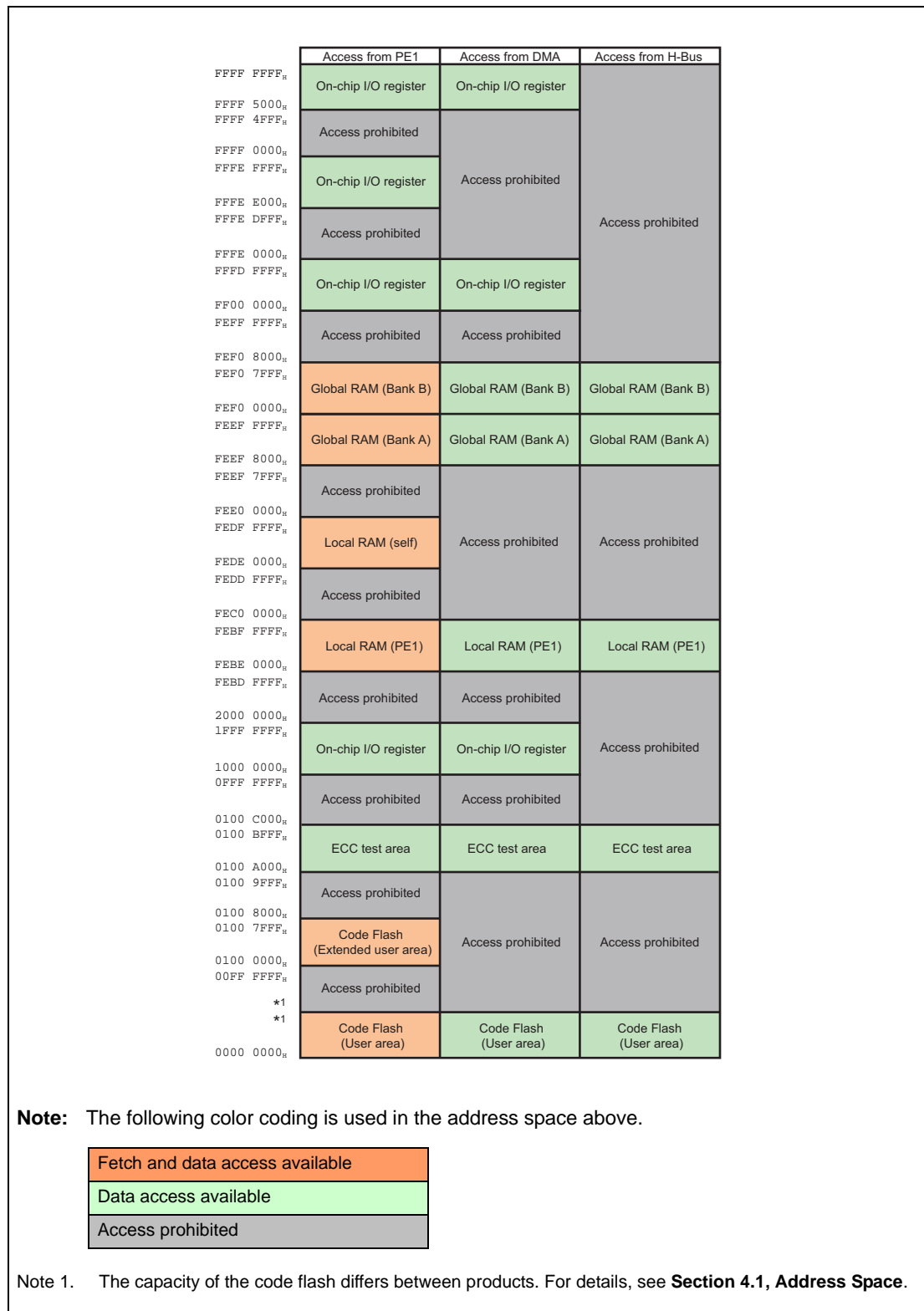


Figure 4.1 Address Space Viewed from Each Bus Master

4.3 Guard Function

The following table shows the guard function for each of memory area. In addition to these guard functions, each PE has MPU function in itself.

Table 4.2 Guard function

| Memory Area | Access from | Guard function |
|-----------------|---------------------------------|---------------------------------|
| P-Bus Area | All bus masters | P-Bus Guard (PBG) |
| LPB Area | Own PE | Internal Peripheral Guard (IPG) |
| | Other bus masters | PE Guard (PEG) |
| Local RAM Area | All bus masters (except own PE) | PE Guard (PEG) |
| H-Bus Area | All bus masters | H-Bus Guard (HBG) |
| Global RAM Area | All bus masters | GRAM Guard (GRG) |

Section 5 Operating Modes

5.1 Features

The device supports 2 operating modes.

Operating modes are selected by the FLMD0 and FLMD1 pins, and decided when pin reset is released.

Table 5.1 shows the list of the operating modes.

Table 5.1 Selection of Operating Modes

| Pins | | Operating Mode |
|------------------|---------------|-------------------------|
| FLMD0 | FLMD1 (P3_14) | |
| 0 | x | Normal operating mode |
| 1 | 0 | Serial programming mode |
| Other than above | | Settings are prohibited |

5.1.1 Normal Operating Mode

This mode is for execution of the user program. The on-chip debug capabilities also use this mode. After release from the reset state, instruction fetch is carried out from the user area. For reset vector of a CPU, see **Section 35, Flash Memory**.

5.1.2 Serial Programming mode

The dedicated flash memory programmer enables erasing/writing to the flash memory. After release from the reset state, this device boots up from the on-chip boot program and starts connection in the specified transmission method. For details, see **Section 35, Flash Memory**.

5.2 External Input Pins

Table 5.2 shows the list of external input pins. Pin levels latched from the FLMD0 and FLMD1 pins are reflected to the mode register (MODE). FLMD1 can be used as GPIO. The level on the FLMD0 pin can be set to prohibit programming and erasure of the code flash memory. When the FLMD0 pin is at the high level, programming and erasure of the flash memory are permitted; when it is at the low level it is prohibited to program or erase the flash memory. For details, see **Section 35, Flash Memory**.

Table 5.2 External Input Pins

| Pin Name | I/O | Function |
|----------|-------|-------------------------------------|
| FLMD0 | input | Primary operating mode select pin |
| FLMD1 | input | Secondary operating mode select pin |

5.3 Register Description

5.3.1 List of Registers

Table 5.3 Register Configuration

| Register Name | Description | Access Width | Initial Value | Address |
|---------------|---------------|--------------|------------------------|------------------------|
| MODE | Mode Register | 32 | 0000 000X _H | FFF8 0104 _H |

Table 5.4 Register Reset Condition

| Register Name | Reset Condition | | | |
|---------------|-----------------|----------------|----------------|---------------------|
| | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| MODE | — | √*1 | — | — |

Note 1. Pin reset only

5.3.2 MODE — Mode Register

This register indicates the operating mode of the device.

The pin levels to be latched from FLMD0 and FLMD1 pins when pin reset is released, are reflected.

Access: This register can only be read in 32-bit units.

Address: FFF8 0104_H

Value after reset: 0000 000X_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FLMD1 | FLMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 5.5 MODE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | FLMD1 | This bit indicates the level of the latch for the FLMD1 pin 0: Low-level detection 1: High-level detection Note: This bit is always read "0" in normal operating mode. |
| 0 | FLMD0 | This bit indicates the level of the latch for the FLMD0 pin 0: Low-level detection 1: High-level detection |

Section 6 Interrupt

The interrupt controller (INTC) determines priority of interrupt sources and controls interrupt requests to the CPU. The INTC has a register to set priority of each interrupt. Interrupt requests are processed according to the priority set in this register by the user.

6.1 Overview

- Interrupt sources
 - One non-maskable interrupt (FENMI): FE level interrupt from ECM
 - One FE level maskable interrupt (FEINT): Pin NMI, OSTM3 to OSTM7
 - 384 EI level maskable interrupts (EIINT)
 - 32 high-speed interrupts
 - DMAC
 - ECM
 - Window watchdog timer
 - 352 low-speed interrupts
 - Timer
 - Communication
 - External interrupt
 - A/D interrupts
 - DTS
 - Flash

For details of FENMI, FEINT, and EIINT, refer to *the RH850G3M User's Manual: Software*.

- Sixteen interrupt priority levels can be set for the EI level interrupts.
Up to 16 priority levels of the EI level interrupts can be set in EI level interrupt control registers.
- Three detection methods
A method of detecting the pin NMI interrupt and external interrupt can be selected from rising edge, falling edge, and both edges.
- Direct branching method or table referencing method is selectable by setting two types of interrupt handler address setting registers.
- Software interrupts
Interrupts of desired priority can be generated from the program by setting software interrupt registers.

6.1.1 Reset Sources

Please refer to **Section 8, Reset Controller**.

6.2 Register Specifications

The INTC has registers listed in tables below. These registers are mainly used to set interrupt priority and to control detection of external interrupt input signals.

6.2.1 Register Configuration

Table 6.1 Interrupt Control

| Address | Register Symbol | Register Name | R/W | Value after Reset | Access Size |
|--|---------------------|-------------------------------------|-----|--|-------------|
| FFFE EA00 _H - FFFE EA3E _H (EIC0 to EIC31) FFFF B040 _H - FFFF B2FE _H (EIC32 to EIC383) | EICn ^{*1} | EI level interrupt control register | R/W | 008F _H ^{*4} 808F _H ^{*5} | 1/8/16 |
| FFFE EAF0 _H (IMR0) FFFF B404 _H - FFFF B42C _H (IMR1 to IMR11) | IMRn ^{*2} | EI level interrupt mask register | R/W | FFFF FFFF _H | 1/8/16/32 |
| FFFE EB00 _H - FFFE EB7C _H (EIBD0 to EIBD31) FFFF B880 _H - FFFF BDFC _H (EIBD32 to EIBD383) | EIBDn ^{*3} | EI level interrupt bind register | R/W | 0000 0001 _H | 32 |
| FFFE EA78 _H | FNC | FE level NMI control register | R | 0000 _H | 1/8/16 |
| FFFE EA7A _H | FIC | FE level interrupt control register | R | 0000 _H | 1/8/16 |
| FFD6 7000 _H | FEINTF | FEINT factor register | R | 0000 0000 _H | 32 |
| FFD6 7008 _H | FEINTFC | FEINT factor clear register | W | 0000 0000 _H | 32 |

Note 1. n = 0 to 383

Note 2. n = 0 to 11

Note 3. n = 0 to 383

Note 4. Synchronous edge detection

Note 5. High-level detection

Among the registers shown in **Table 6.1**, the EIC0 to 31, IMR0, EIBD0 to 31, FNC, and FIC are located in INTC1 of the CPU-specific Peripheral. Each of these registers only can be accessed from the CPU which includes it. Writing is only possible in supervisor mode (PSW.UM = 0).

Of the registers listed in **Table 6.1**, EIC32 to EIC383, IMR1 to IMR11, and EIBD32 to EIBD383 are located in INTC2, the controller for interrupts from peripheral IP group 0. Writing to these registers is only possible for CPU1 in supervisor mode (PSW.UM = 0).

Table 6.2 Software Interrupts

| Address | Register Symbol | Register Name | R/W | Value after Reset | Access Size |
|------------------------|-----------------|-------------------------------|-----|-------------------|-------------|
| FFC0 0000 _H | SINTR0 | Software interrupt register 0 | R/W | 00 _H | 8 |
| FFC0 0004 _H | SINTR1 | Software interrupt register 1 | R/W | 00 _H | 8 |
| FFC0 0008 _H | SINTR2 | Software interrupt register 2 | R/W | 00 _H | 8 |
| FFC0 000C _H | SINTR3 | Software interrupt register 3 | R/W | 00 _H | 8 |
| FFC0 0010 _H | SINTR4 | Software interrupt register 4 | R/W | 00 _H | 8 |

Table 6.3 Interrupt Merge Function

| Address | Register Symbol | Register Name | R/W | Value after Reset | Access Size |
|------------------------|-----------------|--|-----|------------------------|-------------|
| FFF9 8000 _H | PINT0 | Peripheral interrupt status register 0 | R | 0000 0000 _H | 32 |
| FFF9 8004 _H | PINT1 | Peripheral interrupt status register 1 | R | 0000 0000 _H | 32 |
| FFF9 8008 _H | PINT2 | Peripheral interrupt status register 2 | R | 0000 0000 _H | 32 |
| FFF9 800C _H | PINT3 | Peripheral interrupt status register 3 | R | 0000 0000 _H | 32 |
| FFF9 8010 _H | PINT4 | Peripheral interrupt status register 4 | R | 0000 0000 _H | 32 |
| FFF9 8014 _H | PINT5 | Peripheral interrupt status register 5 | R | 0000 0000 _H | 32 |
| FFF9 8018 _H | PINT6 | Peripheral interrupt status register 6 | R | 0000 0000 _H | 32 |
| FFF9 801C _H | PINT7 | Peripheral interrupt status register 7 | R | 0000 0000 _H | 32 |
| FFF9 8020 _H | PINTCLR0 | Interrupt clear register 0 | W | 0000 0000 _H | 32 |
| FFF9 8024 _H | PINTCLR1 | Interrupt clear register 1 | W | 0000 0000 _H | 32 |
| FFF9 8028 _H | PINTCLR2 | Interrupt clear register 2 | W | 0000 0000 _H | 32 |
| FFF9 802C _H | PINTCLR3 | Interrupt clear register 3 | W | 0000 0000 _H | 32 |
| FFF9 8030 _H | PINTCLR4 | Interrupt clear register 4 | W | 0000 0000 _H | 32 |
| FFF9 8034 _H | PINTCLR5 | Interrupt clear register 5 | W | 0000 0000 _H | 32 |
| FFF9 8038 _H | PINTCLR6 | Interrupt clear register 6 | W | 0000 0000 _H | 32 |
| FFF9 803C _H | PINTCLR7 | Interrupt clear register 7 | W | 0000 0000 _H | 32 |

6.2.2 EIC0 - EIC383 — EI Level Interrupt Control Registers 0 to 383

These registers are used to set interrupt control conditions for each EI level interrupt source, and one register is provided for each source of this type. Writing to these registers requires care when a register is set for detection in synchronous edge detection mode. If 0 is written to the EIRFn bit immediately after a peripheral module generates the corresponding interrupt request, the request may be inadvertently lost. On the other hand, writing 1 to the EIRFn bit immediately after an interrupt is accepted by the CPU may lead to incorrect re-issuing of the request. Writing to any of these registers, therefore, should only proceed while peripheral modules are not generating interrupt requests and the CPU has not accepted any interrupt.

Writing to an EICn register includes writing by using the bit operation instructions (set1, clr1, and not1). The bit operation instructions proceed in the following sequence (1) reading the register, (2) processing the specified bit, and (3) writing the new value back to the register. When a bit operation instruction has a bit other than EIRFn as the operand, the value read in step (1) will be written back in step (3). Accordingly, the issuance of interrupt requests from peripheral modules or the acceptance of an interrupt request by the CPU between steps (1) and (3) may lead to the problems described above. Note that access to bits 15 to 13, 11 to 8, 5, and 4 using bit-manipulation instructions is prohibited.

Access: EICn can be read/written in 16-bit units.
EICnH and EICnL can be read/written in 8- or 1-bit units.

Address: FFFE EA00_H - FFFE EA3E_H (EIC0-31)
FFFF B040_H - FFFF B2FE_H (EIC32-383)

Value after reset: 008F_H (edge detection), 808F_H (high level detection)
This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------|----|----|-------|----|----|---|---|-------|-------|---|---|-------|-------|-------|-------|
| | EICTn | — | — | EIRFn | — | — | — | — | EIMKn | EITBn | — | — | EIP3n | EIP2n | EIP1n | EIP0n |
| Value after reset | *1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R/W | R | R | R | R | R/W | R/W | R | R | R/W | R/W | R/W | R/W |

Note 1. 0: Synchronous edge detection, 1: High-level detection

Table 6.4 EIC0 - EIC383 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 15 | EICTn | Interrupt Channel Type The following values are read according to the interrupt input interface. This is a read-only bit. 0: Synchronous edge detection 1: High-level detection When writing to this bit, write the value after reset. |
| 14, 13 | Reserved | When read, the value after reset is returned. When writing to these bits, write the value after reset. |
| 12 | EIRFn | Interrupt Request Flag Operation varies with the interrupt input interface. 0: No interrupt request (Value after reset) 1: Interrupt request present <ul style="list-style-type: none"> Synchronous edge detection This flag is automatically cleared to 0 when an interrupt request of the self-channel is accepted by the CPU core. This bit can be set or cleared by the software. High level detection This bit cannot be set or cleared by the software. This is a read-only bit. |
| 11 to 8 | Reserved | When read, the value after reset is returned. When writing to these bits, write the value after reset. |

Table 6.4 EIC0 - EIC383 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | EIMKn | <p>Interrupt Mask</p> <p>While this bit is set to 1, interrupt requests set in the interrupt request flag (EIRFn) are masked to inhibit interrupt requests from the channel to the CPU core. Notification of presence of unprocessed interrupts is not made and the PMEI bit in ICSR is not set from channels for which this bit is set to 1. Even when interrupt processing is disabled by the setting of this bit, an input of interrupt signal is not masked and the interrupt request flag is set. Setting this bit is also reflected in the setting of corresponding bit of the interrupt mask register (IMR).</p> <p>0: Interrupt processing is enabled. 1: Interrupt processing is disabled. (Value after reset)</p> |
| 6 | EITBn | <p>Interrupt Vector Method Select</p> <p>0: Direct branching method based on priority 1: Table referencing method</p> |
| 5, 4 | Reserved | When read, the value after reset is returned. When writing to these bits, write the value after reset. |
| 3 to 0 | EIP3n-0n | <p>These bits specify 16 interrupt priority levels (0: highest priority, 15: lowest priority).</p> <p>If two or more EI level interrupt requests are generated simultaneously, a source with higher priority specified by these bits is selected and is sent to the CPU core. If the priority specified by these bits is equal, a source of smaller channel number is selected as fixed priority.</p> |

Note: n = 0 to 383

NOTE

The addresses of the registers corresponding to the interrupt channels that are listed as reserved in **Table 6.11, Interrupt Exception Handler and Priority**, are also reserved. Therefore, access to these addresses is prohibited. Operation cannot be guaranteed if they are accessed.

6.2.3 IMR0 - IMR11 — EI Level Interrupt Mask Registers 0 to 11

These registers are aggregation of the EIMK bit in the EIC register. Setting of the corresponding EIMK bit is reflected in each bit in the IMRn register. Furthermore, setting of the IMRn register is reflected in the corresponding EIMK bit.

Access: IMRm can be read/written in 32-bit units.
IMRmH and IMRmL can be read/written in 16-bit units.
IMRmHH, IMRmHL, IMRmLH, and IMRmLL can be read/written in 8- or 1-bit units.

Address: IMR0: FFFE EAF0_H, IMR1: FFFF B404_H, IMR2: FFFF B408_H, IMR3: FFFF B40C_H,
IMR4: FFFF B410_H, IMR5: FFFF B414_H, IMR6: FFFF B418_H, IMR7: FFFF B41C_H,
IMR8: FFFF B420_H, IMR9: FFFF B424_H, IMR10: FFFF B428_H, IMR11: FFFF B42C_H

Value after reset: FFFF FFFF_H This register is initialized by any reset.

IMR0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| IMR0H | EIMK31 | EIMK30 | EIMK29 | EIMK28 | EIMK27 | EIMK26 | EIMK25 | EIMK24 | EIMK23 | EIMK22 | EIMK21 | EIMK20 | EIMK19 | EIMK18 | EIMK17 | EIMK16 |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IMR0L | EIMK15 | EIMK14 | EIMK13 | EIMK12 | EIMK11 | EIMK10 | EIMK9 | EIMK8 | EIMK7 | EIMK6 | EIMK5 | EIMK4 | EIMK3 | EIMK2 | EIMK1 | EIMK0 |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

IMR1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| IMR1H | EIMK63 | EIMK62 | EIMK61 | EIMK60 | EIMK59 | EIMK58 | EIMK57 | EIMK56 | EIMK55 | EIMK54 | EIMK53 | EIMK52 | EIMK51 | EIMK50 | EIMK49 | EIMK48 |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IMR1L | EIMK47 | EIMK46 | EIMK45 | EIMK44 | EIMK43 | EIMK42 | EIMK41 | EIMK40 | EIMK39 | EIMK38 | EIMK37 | EIMK36 | EIMK35 | EIMK34 | EIMK33 | EIMK32 |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

:

:

IMR11

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| IMR11H | EIMK383 | EIMK382 | EIMK381 | EIMK380 | EIMK379 | EIMK378 | EIMK377 | EIMK376 | EIMK375 | EIMK374 | EIMK373 | EIMK372 | EIMK371 | EIMK370 | EIMK369 | EIMK368 |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| IMR11L | EIMK 367 | EIMK 366 | EIMK 365 | EIMK 364 | EIMK 363 | EIMK 362 | EIMK 361 | EIMK 360 | EIMK 359 | EIMK 358 | EIMK 357 | EIMK 356 | EIMK 355 | EIMK 354 | EIMK 353 | EIMK 352 |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

NOTE

The bits corresponding to the interrupt channels that are listed as reserved in **Table 6.11, Interrupt Exception Handler and Priority**, are also reserved. When read, the value after reset is read. When writing, write the value after reset.

6.2.4 EIBD0 - EIBD383 — EI Level Interrupt Bind Registers 0 to 383

These registers are provided for each EI level interrupt source to make correspondence between each source and PE.

Access: These registers can be read/written in 32-bit units.

Address: FFFE EB00_H - FFFE EB7C_H (EIBD0 to EIBD31)
FFFF B880_H - FFFF BDFC_H (EIBD32 to EIBD383)

Value after reset: 0000 0001_H This register is initialized by any reset.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | GPID[1:0] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PEID[2:0] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 6.5 EIBD0 - EIBD383 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 18 | Reserved | When read, the value after reset is returned. When writing to these bits, write the value after reset. |
| 17, 16 | GPID | These bits are provided only in EIBD32 to EIBD383. Be sure to set these bits to 00 in this product. These bits are reserved for EIBD0 to EIBD31. When read, the value after reset is returned. When writing to these bits, write the value after reset. |
| 15 to 3 | Reserved | When read, the value after reset is returned. When writing to these bits, write the value after reset. |
| 2 to 0 | PEID | These bits specify an interrupt bind (request) destination. These bits are fixed to 001 in EIBD0 to EIBD31 and cannot be modified. Be sure to set these bits in EIBD32 to EIBD383 to 001 in this product. |

NOTE

The addresses of the registers corresponding to the interrupt channels that are listed as reserved in **Table 6.11, Interrupt Exception Handler and Priority**, are also reserved. Therefore, access to these addresses is prohibited. Operation cannot be guaranteed if they are accessed.

CAUTION

Changing the corresponding EIBDn register during the processing of an EIINT request is prohibited.

6.2.5 FNC — FE Level NMI Control Register

This register is used to set FE level NMI control conditions.

Access: FNC can be read in 16-bit units.
FNCH and FNCL can be read in 8- or 1-bit units.

Address: FFFE EA78_H

Value after reset: 0000_H (synchronous edge detection), This register is initialized by any reset.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|------------------|----|----|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | FNR _F | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 6.6 FNC Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 15 to 13 | Reserved | When read, the value after reset is returned. |
| 12 | FNR _F | Interrupt Request Flag 0: No interrupt request (Value after reset) 1: An interrupt request present This bit is automatically cleared to 0 when an FE level NMI interrupt request is accepted by the CPU core. |
| 11 to 0 | Reserved | When read, the value after reset is returned. |

6.2.6 FIC — FE Level Interrupt Control Register

This register is used to set FE level interrupt control conditions.

Access: FIC can be read in 16-bit units.
FICH and FICL can be read in 8- or 1-bit units.

Address: FFFE EA7A_H

Value after reset: 0000_H (synchronous edge detection), This register is initialized by any reset.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|------------------|----|----|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | FIR _F | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 6.7 FIC Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 15 to 13 | Reserved | When read, the value after reset is returned. |
| 12 | FIR _F | Interrupt Request Flag 0: No interrupt request (Value after reset) 1: An interrupt request present This bit is automatically cleared to 0 when an FE level interrupt request is accepted by the CPU core. |
| 11 to 0 | Reserved | When read, the value after reset is returned. |

6.2.7 SINTR0 - SINTR4 — Software Interrupt Registers

SINTR0 to SINTR4 are 8-bit registers to control software interrupts 0 to 4 (SINT0 to SINT4).

Writing 01_H to any of these registers increments the counter value by 1. Writing 00_H to any of the registers decrements the counter value by 1. When the values of these registers are 1 or larger, a software interrupt (SINT0 to SINT4) is generated. The value read from any of these registers is the current value of the register.

Access: These registers can be read/written in 8- or 1-bit units.

Address: SINT0: FFC0 0000_H, SINT1: FFC0 0004_H, SINT2: FFC0 0008_H,
SINT3: FFC0 000C_H, SINT4: FFC0 0010_H

Value after reset: 00_H This register is initialized by any reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| | SINTCn7 | SINTCn6 | SINTCn5 | SINTCn4 | SINTCn3 | SINTCn2 | SINTCn1 | SINTCn0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 6.8 SINTR0 - SINTR4 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 7 to 0 | SINTCn[7:0] | Software Interrupt Request These bits generate a software interrupt. [Reading operation] The counter value of the SINTn interrupt request count is read. [Writing operation] Writing 01 _H : The counter value is incremented.*1 Writing 00 _H : The counter value is decremented.*2 Other than the above: Forbidden (No guarantee of operation) |

Note 1. When 01_H is written while the value of the counter is FF_H, the value is not incremented, but remains at FF_H.

Note 2. When 00_H is written while the value of the counter is 00_H, the value is not decremented, but remains at 00_H.

6.2.8 PINT0 — PINT7 — Peripheral Interrupt Status Registers PINTCLR0 - PINTCLR7 — Peripheral Interrupt Status Clear Registers

The DTS transfer completion interrupt and transfer count match interrupt are merged in 32-channel units. PINT0 to PINT7 contain the interrupt status flags to indicate the originating channels for these interrupts.

Writing the interrupt status register value to the interrupt status clear register (PINTCLR0 to PINTCLR7) of the same channel in the interrupt handler clears interrupts. Interrupts on the lower-bits side take precedence to support multiplex interrupts.

PINT0 to PINT7 are initialized to 0000 0000_H by any reset source.

Access: PINT_n can be read in 32-bit units.

PINTCLR_n can be written in 32-bit units.

Address: PINT0: FFF9 8000_H, PINT1: FFF9 8004_H, PINT2: FFF9 8008_H, PINT3: FFF9 800C_H
PINT4: FFF9 8010_H, PINT5: FFF9 8014_H, PINT6: FFF9 8018_H, PINT7: FFF9 801C_H
PINTCLR0: FFF9 8020_H, PINTCLR1: FFF9 8024_H
PINTCLR2: FFF9 8028_H, PINTCLR3: FFF9 802C_H
PINTCLR4: FFF9 8030_H, PINTCLR5: FFF9 8034_H
PINTCLR6: FFF9 8038_H, PINTCLR7: FFF9 803C_H

Value after reset: 0000 0000_H

PINT n + x (n = 0 to 3, x = 0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|-------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | INTDTS [31+32*n] | INTDTS [30+32*n] | INTDTS [29+32*n] | INTDTS [28+32*n] | INTDTS [27+32*n] | INTDTS [26+32*n] | INTDTS [25+32*n] | INTDTS [24+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | INTDTS [23+32*n] | INTDTS [22+32*n] | INTDTS [21+32*n] | INTDTS [20+32*n] | INTDTS [19+32*n] | INTDTS [18+32*n] | INTDTS [17+32*n] | INTDTS [16+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | INTDTS [15+32*n] | INTDTS [14+32*n] | INTDTS [13+32*n] | INTDTS [12+32*n] | INTDTS [11+32*n] | INTDTS [10+32*n] | INTDTS [9+32*n] | INTDTS [8+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | INTDTS [7+32*n] | INTDTS [6+32*n] | INTDTS [5+32*n] | INTDTS [4+32*n] | INTDTS [3+32*n] | INTDTS [2+32*n] | INTDTS [1+32*n] | INTDTS [0+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

PINT n + x (n = 0 to 3, x = 4)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | INTCTDTS [31+32*n] | INTCTDTS [30+32*n] | INTCTDTS [29+32*n] | INTCTDTS [28+32*n] | INTCTDTS [27+32*n] | INTCTDTS [26+32*n] | INTCTDTS [25+32*n] | INTCTDTS [24+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | INTCTDTS [23+32*n] | INTCTDTS [22+32*n] | INTCTDTS [21+32*n] | INTCTDTS [20+32*n] | INTCTDTS [19+32*n] | INTCTDTS [18+32*n] | INTCTDTS [17+32*n] | INTCTDTS [16+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------|----------------------|
| | INTCTDTS [15+32*n] | INTCTDTS [14+32*n] | INTCTDTS [13+32*n] | INTCTDTS [12+32*n] | INTCTDTS [11+32*n] | INTCTDTS [10+32*n] | INTCTDTS [9+32*n] | INTCTDTS [8+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| | INTCTDTS [7+32*n] | INTCTDTS [6+32*n] | INTCTDTS [5+32*n] | INTCTDTS [4+32*n] | INTCTDTS [3+32*n] | INTCTDTS [2+32*n] | INTCTDTS [1+32*n] | INTCTDTS [0+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

PINTCLR n + x (n = 0 to 3, x = 0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|-------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | INTCLR [31+32*n] | INTCLR [30+32*n] | INTCLR [29+32*n] | INTCLR [28+32*n] | INTCLR [27+32*n] | INTCLR [26+32*n] | INTCLR [25+32*n] | INTCLR [24+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | INTCLR [23+32*n] | INTCLR [22+32*n] | INTCLR [21+32*n] | INTCLR [20+32*n] | INTCLR [19+32*n] | INTCLR [18+32*n] | INTCLR [17+32*n] | INTCLR [16+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|-------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|--------------------|--------------------|
| | INTCLR [15+32*n] | INTCLR [14+32*n] | INTCLR [13+32*n] | INTCLR [12+32*n] | INTCLR [11+32*n] | INTCLR [10+32*n] | INTCLR [9+32*n] | INTCLR [8+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | INTCLR [7+32*n] | INTCLR [6+32*n] | INTCLR [5+32*n] | INTCLR [4+32*n] | INTCLR [3+32*n] | INTCLR [2+32*n] | INTCLR [1+32*n] | INTCLR [0+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |

PINTCLR n + x (n = 0 to 3, x = 4)

| | | | | | | | | |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| | INTCTCLR [31+32*n] | INTCTCLR [30+32*n] | INTCTCLR [29+32*n] | INTCTCLR [28+32*n] | INTCTCLR [27+32*n] | INTCTCLR [26+32*n] | INTCTCLR [25+32*n] | INTCTCLR [24+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | INTCTCLR [23+32*n] | INTCTCLR [22+32*n] | INTCTCLR [21+32*n] | INTCTCLR [20+32*n] | INTCTCLR [19+32*n] | INTCTCLR [18+32*n] | INTCTCLR [17+32*n] | INTCTCLR [16+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | INTCTCLR [15+32*n] | INTCTCLR [14+32*n] | INTCTCLR [13+32*n] | INTCTCLR [12+32*n] | INTCTCLR [11+32*n] | INTCTCLR [10+32*n] | INTCTCLR [9+32*n] | INTCTCLR [8+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | INTCTCLR [7+32*n] | INTCTCLR [6+32*n] | INTCTCLR [5+32*n] | INTCTCLR [4+32*n] | INTCTCLR [3+32*n] | INTCTCLR [2+32*n] | INTCTCLR [1+32*n] | INTCTCLR [0+32*n] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |

PINT0

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 0 | INTDTS[31:0] | DTS transfer completion interrupt status on ch31 to ch0 |

PINT1

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 31 to 0 | INTDTS[63:32] | DTS transfer completion interrupt status on ch63 to ch32 |

PINT2

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 31 to 0 | INTDTS[95:64] | DTS transfer completion interrupt status on ch95 to ch64 |

PINT3

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 31 to 0 | INTDTS[127:96] | DTS transfer completion interrupt status on ch127 to ch96 |

PINT4

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 31 to 0 | INTCTDTS[31:0] | DTS transfer count match interrupt status on ch31 to ch0 |

PINT5

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 0 | INTCTDTS [63:32] | DTS transfer count match interrupt status on ch63 to ch32 |

PINT6

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 0 | INTCTDTS [95:64] | DTS transfer count match interrupt status on ch95 to ch64 |

PINT7

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 31 to 0 | INTCTDTS [127:96] | DTS transfer count match interrupt status on ch127 to ch96 |

PINTCLR0

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 0 | INTCLR[31:0] | These bits clear the DTS transfer completion interrupt status on ch31 to ch0. Within the interrupt handler, these bits write the value read from PINT0. |

PINTCLR1

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 31 to 0 | INTCLR[63:32] | These bits clear the DTS transfer completion interrupt status on ch63 to ch32. Within the interrupt handler, these bits write the value read from PINT1. |

PINTCLR2

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 31 to 0 | INTCLR[95:64] | These bits clear the DTS transfer completion interrupt status on ch95 to ch64. Within the interrupt handler, these bits write the value read from PINT2. |

PINTCLR3

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 31 to 0 | INTCLR[127:96] | These bits clear the DTS transfer completion interrupt status on ch127 to ch96. Within the interrupt handler, these bits write the value read from PINT3. |

PINTCLR4

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 31 to 0 | INTCTCLR[31:0] | These bits clear the DTS transfer count match interrupt status on ch31 to ch0. Within the interrupt handler, these bits write the value read from PINT4. |

PINTCLR5

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 0 | INTCTCLR [63:32] | These bits clear the DTS transfer count match interrupt status on ch63 to ch32. Within the interrupt handler, these bits write the value read from PINT5. |

PINTCLR6

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 0 | INTCTCLR [95:64] | These bits clear the DTS transfer count match interrupt status on ch95 to ch64. Within the interrupt handler, these bits write the value read from PINT6. |

PINTCLR7

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 31 to 0 | INTCTCLR [127:96] | These bits clear the DTS transfer count match interrupt status on ch127 to ch96. Within the interrupt handler, these bits write the value read from PINT7. |

CAUTIONS

1. Peripheral Interrupt Status Clear Registers (PINTCLR0 - PINTCLR7) are always read as 0.
2. There is some delay between writing the INTCLRn (n=0 to 127) bit or INTCTCLRn (n=0 to 127) bit and clearing the INTDTSn (n=0 to 127) flag or the INTCTDTSn (n=0 to 127) flag. It is recommended to read Peripheral Interrupt Status Registers (PINT0 to PINT7) twice after clearing one of its bits. The second read operation will read the correct status.

6.2.9 FEINTF — FEINT Factor Register

The FE level interrupt (FEINT) factor register is used in common by the OSTM3 to OSTM7 interrupts and the pin NMI interrupt. When an FE level interrupt occurs, check the interrupt source by referring to this register.

Access: This register can be read in 32-bit units.

Address: FFD6 7000_H

Value after reset: 0000 0000_H This register is initialized by any reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|------------------|------------------|------------------|------------------|------------------|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | INTOST M7FEIF | INTOST M6FEIF | INTOST M5FEIF | INTOST M4FEIF | INTOST M3FEIF | NMIFEI F |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 6.9 FEINTF Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 6 | Reserved | When read, the value after reset is returned. |
| 5 | INTOSTM7FEIF | OSTM7 Interrupt Generation 0: OSTM7 interrupt has not been generated. 1: OSTM7 interrupt has been generated. |
| 4 | INTOSTM6FEIF | OSTM6 Interrupt Generation 0: OSTM6 interrupt has not been generated. 1: OSTM6 interrupt has been generated. |
| 3 | INTOSTM5FEIF | OSTM5 Interrupt Generation 0: OSTM5 interrupt has not been generated. 1: OSTM5 interrupt has been generated. |
| 2 | INTOSTM4FEIF | OSTM4 Interrupt Generation 0: OSTM4 interrupt has not been generated. 1: OSTM4 interrupt has been generated. |
| 1 | INTOSTM3FEIF | OSTM3 Interrupt Generation 0: OSTM3 interrupt has not been generated. 1: OSTM3 interrupt has been generated. |
| 0 | NMIFEIF | Pin NMI Interrupt Generation 0: Pin NMI interrupt has not been generated. 1: Pin NMI interrupt has been generated. |

6.2.10 FEINTFC — FEINT Factor Clear Register

Access: This register can be written in 32-bit units.

Address: FFD6 7008_H

Value after reset: 0000 0000_H This register is initialized by any reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----------------|----------------|----------------|----------------|----------------|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | OSTM7 FEIFC | OSTM6 FEIFC | OSTM5 FEIFC | OSTM4 FEIFC | OSTM3 FEIFC | NMIFEI FC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | W | W | W | W | W | W |

Table 6.10 FEINTFC Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 6 | Reserved | When writing, always write 0. |
| 5 | OSTM7FEIFC | INTOSTM7FEIF Clear 0: The INTOSTM7FEIF bit is not cleared. 1: The INTOSTM7FEIF bit is cleared. |
| 4 | OSTM6FEIFC | INTOSTM6FEIF Clear 0: The INTOSTM6FEIF bit is not cleared. 1: The INTOSTM6FEIF bit is cleared. |
| 3 | OSTM5FEIFC | INTOSTM5FEIF Clear 0: The INTOSTM5FEIF bit is not cleared. 1: The INTOSTM5FEIF bit is cleared. |
| 2 | OSTM4FEIFC | INTOSTM4FEIF Clear 0: The INTOSTM4FEIF bit is not cleared. 1: The INTOSTM4FEIF bit is cleared. |
| 1 | OSTM3FEIFC | INTOSTM3FEIF Clear 0: The INTOSTM3FEIF bit is not cleared. 1: The INTOSTM3FEIF bit is cleared. |
| 0 | NMIFEIFC | NMIFEIF Clear 0: The NMIFEIF bit is not cleared. 1: The NMIFEIF bit is cleared. |

6.3 Interrupt Sources

There are five groups of interrupt sources: NMI, INTP_n, ECM interrupt, software interrupts (SINT), and interrupts from the on-chip peripheral modules. The priority level of each interrupt is represented by an interrupt priority value between 0 (the highest level) and 15 (the lowest level).

6.3.1 NMI Interrupts

NMI interrupts are input from the NMI pin. A method of detecting the NMI interrupt can be selected from rising edge, falling edge, and both edges. For details on the detection method, see **Section 2, Pin Functions**.

6.3.2 INTP_n Interrupts

INTP_n interrupts are input from INTP_n pins. A method of detecting the INTP_n interrupt can be selected from rising edge, falling edge, and both edges. For details on the detection method, see **Section 2, Pin Functions**.

Furthermore, interrupt priority levels 0 to 15 can be set for each pin in the interrupt control register (EIC_m).

When an interrupt request is detected by a change in the INTP_n pin, and an interrupt request signal is sent to the INTC. The INTP_n interrupt request detection result is retained until the interrupt request is accepted. Whether an INTP_n interrupt request has been detected or not can be checked by reading the EIRF_n bit in the corresponding EI level interrupt control register *n* (EIC_n) for INTP_n. Writing 0 clears the INTP_n interrupt request detection result.

When exiting the INTP_n interrupt exception handler, confirm that the EIRF_n bit has been cleared in the corresponding EI level interrupt control register *n* (EIC_n) for INTP_n to prevent re-acceptance by mistake, and then issue an instruction to return from interrupt.

6.3.3 ECM Interrupts

The Error Control Module (ECM) generates ECM interrupt requests by merging error interrupts from multiple sources. For details, see **Section 32, Error Control Module (ECM)** and **Section 31, Functional Safety**.

6.3.4 Software Interrupts

Software interrupt (SINT) priority can be set within priority levels 0 to 15 for each interrupt source.

6.3.5 On-Chip Peripheral Module Interrupts

For the on-chip peripheral module that generates an interrupt, see **Table 6.11, Interrupt Exception Handler and Priority**.

Since different interrupt vectors are assigned to each source, the interrupt exception handler need not decide sources. Priority can be set within priority levels 0 to 15 for each interrupt source.

6.4 Interrupt Exception Handler and Priority Operations

Table 6.11 lists interrupt sources, source codes, exception handler offset addresses, and interrupt priority.

There are two specifications for exception handler addresses: (1) standard specifications where exception handler addresses are determined by the PSW.EBV bit in the CPU core, RBASE register, and EBASE register and (2) extended specifications where exception handler addresses for interrupts are specified individually for each channel.

In the standard specifications, an offset address is added to the base address (RBASE register /EBASE register) in the CPU core to generate an exception handler address. The following two methods are provided for giving an interrupt offset address. For channels other than the interrupt channel, the specified offset address is given.

- An offset address is determined within a range of +100H to +1F0H according to the priority level (0 to 15) specified for each channel, regardless of interrupt channels (offset address, direct branch, RINT = 0 in **Table 6.11**).
- Every offset address is +100H regardless of the priority level. This is a function to reduce the memory occupation size of the exception handler (offset address, direct branch, RINT = 1 in **Table 6.11**).

In the extended specifications, a table for reading exception handler addresses for each interrupt channel is provided. The handler address is extracted by referencing the table. The table reference position is obtained by the following calculation formula (offset address, table reference in **Table 6.11**). The INTBP register exists in the CPU core.

$$\text{Exception handler address read position} = \text{INTBP} + \text{channel number} * 4\text{-byte}$$

For details of handler addresses, see *the RH850G3M User's Manual: Software*.

Priority levels 0 to 15 (0: highest) of INTP_n (external interrupts) and on-chip peripheral module interrupts can be set for each channel. If specified priority levels are equal, an interrupt source of smaller channel number is selected as fixed priority.

CAUTION

FENMI, FEINT, EIINT (direct vector method), SYSERR, FPI need insertion of SYNC instruction before the exception handler.

For details, see. the *RH850G3M User's Manual: Software*.

Table 6.11 Interrupt Exception Handler and Priority (1/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|------------------------|--|------------------------------|-------------------------------------|-----------------|---|---|-------------------|--|------------------|------|------|
| | | | | | Direct Branch | | Table Reference | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| Non-maskable interrupt | Error control module (non-maskable) | | (FENMI) | E0 _H | +0E0 _H | +0E0 _H | — | | High | √ | √ |
| FE level interrupt | OSTM3 to OSTM7 NMI pin interrupt | | (FEINT) | F0 _H | +0F0 _H | +0F0 _H | — | | | √ | √ |
| | Reserved | | 0 | 1000 | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +000 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 1 | 1001 | | | +004 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 2 | 1002 | | | +008 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 3 | 1003 | | | +00C _H | 0 to 15 (15) | | — | — |
| | Reserved | | 4 | 1004 | | | +010 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 5 | 1005 | | | +014 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 6 | 1006 | | | +018 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 7 | 1007 | | | +01C _H | 0 to 15 (15) | | — | — |
| Error control module | Error control module (maskable) (INTECM) | | 8 | 1008 | | | +020 _H | 0 to 15 (15) | | √ | √ |
| WDTA | WDTATIT 75% interrupt (INTWDTA0) | | 9 | 1009 | | | +024 _H | 0 to 15 (15) | | √ | √ |
| | Reserved | | 10 | 100A | | | +028 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 11 | 100B | | | +02C _H | 0 to 15 (15) | | — | — |
| | Reserved | | 12 | 100C | | | +030 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 13 | 100D | | | +034 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 14 | 100E | | | +038 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 15 | 100F | | | +03C _H | 0 to 15 (15) | | — | — |
| DMA | DMA0 transfer end/count match (INTDMA0) | | 16 | 1010 | | | +040 _H | 0 to 15 (15) | | √ | √ |
| | DMA1 transfer end/count match (INTDMA1) | | 17 | 1011 | | | +044 _H | 0 to 15 (15) | | √ | √ |
| | DMA2 transfer end/count match (INTDMA2) | | 18 | 1012 | | | +048 _H | 0 to 15 (15) | | √ | √ |
| | DMA3 transfer end/count match (INTDMA3) | | 19 | 1013 | | | +04C _H | 0 to 15 (15) | | √ | √ |
| | DMA4 transfer end/count match (INTDMA4) | | 20 | 1014 | | | +050 _H | 0 to 15 (15) | | √ | √ |
| | DMA5 transfer end/count match (INTDMA5) | | 21 | 1015 | | | +054 _H | 0 to 15 (15) | | √ | √ |
| | DMA6 transfer end/count match (INTDMA6) | | 22 | 1016 | | | +058 _H | 0 to 15 (15) | | √ | √ |
| | DMA7 transfer end/count match (INTDMA7) | | 23 | 1017 | | | +05C _H | 0 to 15 (15) | | √ | √ |
| | DMA8 transfer end/count match (INTDMA8) | | 24 | 1018 | | | +060 _H | 0 to 15 (15) | | √ | √ |
| | DMA9 transfer end/count match (INTDMA9) | | 25 | 1019 | | | +064 _H | 0 to 15 (15) | | √ | √ |
| | DMA10 transfer end/count match (INTDMA10) | | 26 | 101A | | | +068 _H | 0 to 15 (15) | | √ | √ |
| | DMA11 transfer end/count match (INTDMA11) | | 27 | 101B | | | +06C _H | 0 to 15 (15) | | √ | √ |
| | DMA12 transfer end/count match (INTDMA12) | | 28 | 101C | | | +070 _H | 0 to 15 (15) | | √ | √ |
| | DMA13 transfer end/count match (INTDMA13) | | 29 | 101D | | | +074 _H | 0 to 15 (15) | | √ | √ |
| | DMA14 transfer end/count match (INTDMA14) | | 30 | 101E | | | +078 _H | 0 to 15 (15) | | √ | √ |
| | DMA15 transfer end/count match (INTDMA15) | | 31 | 101F | | | +07C _H | 0 to 15 (15) | | √ | √ |
| INTP | External interrupt 0 (INTP0) | | 32 | 1020 | | | +080 _H | 0 to 15 (15) | | √ | √ |
| | External interrupt 1 (INTP1) | | 33 | 1021 | | | +084 _H | 0 to 15 (15) | | √ | √ |
| | External interrupt 2 (INTP2) | | 34 | 1022 | | | +088 _H | 0 to 15 (15) | | √ | √ |
| | External interrupt 3 (INTP3) | | 35 | 1023 | | | +08C _H | 0 to 15 (15) | | √ | √ |
| | External interrupt 4 (INTP4) | | 36 | 1024 | | | +090 _H | 0 to 15 (15) | | √ | √ |
| | Reserved | | 37 | 1025 | | | +094 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 38 | 1026 | | | +098 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 39 | 1027 | | | +09C _H | 0 to 15 (15) | | — | — |
| TSG30 | TSG30 compare match interrupt 0 (INTTSG30I0) | | 40 | 1028 | | | +0A0 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 1 (INTTSG30I1) | | 41 | 1029 | | | +0A4 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 2 (INTTSG30I2) | | 42 | 102A | | | +0A8 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 3 (INTTSG30I3) | | 43 | 102B | | | +0AC _H | 0 to 15 (15) | | √ | √ |

Table 6.11 Interrupt Exception Handler and Priority (2/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|--|--|--|-------------------------------------|-------------|---|---|-------------------|--|------------------|--------------|------|
| | | | | | Direct Branch | | Table Reference | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| TSG30 | TSG30 compare match interrupt 4 (INTTSG30I4) | | 44 | 102C | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +0B0 _H | 0 to 15 (15) | High | √ | √ |
| | TSG30 compare match interrupt 5 (INTTSG30I5) | | 45 | 102D | | | +0B4 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 6 (INTTSG30I6) | | 46 | 102E | | | +0B8 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 7 (INTTSG30I7) | | 47 | 102F | | | +0BC _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 8 (INTTSG30I8) | | 48 | 1030 | | | +0C0 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 9 (INTTSG30I9) | | 49 | 1031 | | | +0C4 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 10 (INTTSG30I10) | | 50 | 1032 | | | +0C8 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 11 (INTTSG30I11) | | 51 | 1033 | | | +0CC _H | 0 to 15 (15) | | √ | √ |
| | TSG30 compare match interrupt 12 (INTTSG30I12) | | 52 | 1034 | | | +0D0 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 peak interrupt (INTTSG30IPEK) | | 53 | 1035 | | | +0D4 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 valley interrupt (INTTSG30IVLY) | | 54 | 1036 | | | +0D8 _H | 0 to 15 (15) | | √ | √ |
| | TSG30 timer error interrupt (INTTSG30IER) | | 55 | 1037 | | | +0DC _H | 0 to 15 (15) | | √ | √ |
| | TSG30 timer warning interrupt (INTTSG30IWN) | | 56 | 1038 | | | +0E0 _H | 0 to 15 (15) | | √ | √ |
| | TSG31 | TSG31 compare match interrupt 0 (INTTSG31I0) | | 57 | | | 1039 | +0E4 _H | | 0 to 15 (15) | √ |
| TSG31 compare match interrupt 1 (INTTSG31I1) | | | 58 | 103A | +0E8 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 2 (INTTSG31I2) | | | 59 | 103B | +0EC _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 3 (INTTSG31I3) | | | 60 | 103C | +0F0 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 4 (INTTSG31I4) | | | 61 | 103D | +0F4 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 5 (INTTSG31I5) | | | 62 | 103E | +0F8 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 6 (INTTSG31I6) | | | 63 | 103F | +0FC _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 7 (INTTSG31I7) | | | 64 | 1040 | +100 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 8 (INTTSG31I8) | | | 65 | 1041 | +104 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 9 (INTTSG31I9) | | | 66 | 1042 | +108 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 10 (INTTSG31I10) | | | 67 | 1043 | +10C _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 11 (INTTSG31I11) | | | 68 | 1044 | +110 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 compare match interrupt 12 (INTTSG31I12) | | | 69 | 1045 | +114 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 peak interrupt (INTTSG31IPEK) | | | 70 | 1046 | +118 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 valley interrupt (INTTSG31IVLY) | | | 71 | 1047 | +11C _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 timer error interrupt (INTTSG31IER) | | | 72 | 1048 | +120 _H | 0 to 15 (15) | √ | √ | | | |
| TSG31 timer warning interrupt (INTTSG31IWN) | | | 73 | 1049 | +124 _H | 0 to 15 (15) | √ | √ | | | |
| OSTM0 | | OSTM0 interrupt (INTOSTM0) | | 74 | 104A | +128 _H | 0 to 15 (15) | √ | √ | | |
| OSTM1 | | OSTM1 interrupt (INTOSTM1) | | 75 | 104B | +12C _H | 0 to 15 (15) | √ | √ | | |
| ADCG0 | ADCG0 error interrupt (INTADCG0ERR) | | 76 | 104C | +130 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG0 SG0 end interrupt (INTADCG0I0) | | 77 | 104D | +134 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG0 SG1 end interrupt (INTADCG0I1) | | 78 | 104E | +138 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG0 SG2 end interrupt (INTADCG0I2) | | 79 | 104F | +13C _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG0 SG3 end interrupt (INTADCG0I3) | | 80 | 1050 | +140 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG0 SG4 end interrupt (INTADCG0I4) | | 81 | 1051 | +144 _H | 0 to 15 (15) | √ | √ | | | |
| CSIH0 | CSIH0 communication error interrupt (INTCSIH0IRE) | | 82 | 1052 | +148 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH0 receive status/CS0 receive status interrupt (INTCSIH0IR0S) | | 83 | 1053 | +14C _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH0 communication status/CS0 communication status interrupt (INTCSIH0IC0S) | | 84 | 1054 | +150 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH0 CS1 receive status interrupt (INTCSIH0IR1) | | 85 | 1055 | +154 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH0 CS1 communication status interrupt (INTCSIH0IC1) | | 86 | 1056 | +158 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH0 CS2 receive status interrupt (INTCSIH0IR2) | | 87 | 1057 | +15C _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH0 CS2 communication status interrupt (INTCSIH0IC2) | | 88 | 1058 | +160 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH0 JOB end interrupt (INTCSIH0IJC) | | 89 | 1059 | +164 _H | 0 to 15 (15) | √ | √ | | | |

Table 6.11 Interrupt Exception Handler and Priority (3/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | Table Reference | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|-------------------|---|------------------------------|-------------------------------------|-------------|---|---|-------------------|--|------------------|------|------|
| | | | | | Direct Branch | | | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| CSIH1 | CSIH1 communication error interrupt (INTCSIH1IRE) | | 90 | 105A | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +168 _H | 0 to 15 (15) | High | √ | √ |
| | CSIH1 receive status/CS0 receive status interrupt (INTCSIH1R0S) | | 91 | 105B | | | +16C _H | 0 to 15 (15) | | √ | √ |
| | CSIH1 communication status/CS0 communication status interrupt (INTCSIH1C0S) | | 92 | 105C | | | +170 _H | 0 to 15 (15) | | √ | √ |
| | CSIH1 CS1 receive status interrupt (INTCSIH1R1) | | 93 | 105D | | | +174 _H | 0 to 15 (15) | | √ | √ |
| | CSIH1 CS1 communication status interrupt (INTCSIH1C1) | | 94 | 105E | | | +178 _H | 0 to 15 (15) | | √ | √ |
| | CSIH1 CS2 receive status interrupt (INTCSIH1R2) | | 95 | 105F | | | +17C _H | 0 to 15 (15) | | √ | √ |
| | CSIH1 CS2 communication status interrupt (INTCSIH1C2) | | 96 | 1060 | | | +180 _H | 0 to 15 (15) | | √ | √ |
| | CSIH1 JOB end interrupt (INTCSIH1JJC) | | 97 | 1061 | | | +184 _H | 0 to 15 (15) | | √ | √ |
| CSIH2 | CSIH2 communication error interrupt (INTCSIH2IRE) | | 98 | 1062 | +188 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH2 receive status interrupt (INTCSIH2IR) | | 99 | 1063 | +18C _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH2 communication status interrupt (INTCSIH2IC) | | 100 | 1064 | +190 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH2 JOB end interrupt (INTCSIH2JJC) | | 101 | 1065 | +194 _H | 0 to 15 (15) | √ | √ | | | |
| CSIH3 | CSIH3 communication error interrupt (INTCSIH3IRE) | | 102 | 1066 | +198 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH3 receive status interrupt (INTCSIH3IR) | | 103 | 1067 | +19C _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH3 communication status interrupt (INTCSIH3IC) | | 104 | 1068 | +1A0 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIH3 JOB end interrupt (INTCSIH3JJC) | | 105 | 1069 | +1A4 _H | 0 to 15 (15) | √ | √ | | | |
| SCI30 | SCI30 receive error (INTSCI30ERI) | √ | 106 | 106A | +1A8 _H | 0 to 15 (15) | √ | √ | | | |
| | SCI30 receive data full (INTSCI30RXI) | | 107 | 106B | +1AC _H | 0 to 15 (15) | √ | √ | | | |
| | SCI30 transmit data empty (INTSCI30TXI) | | 108 | 106C | +1B0 _H | 0 to 15 (15) | √ | √ | | | |
| | SCI30 transmit end (INTSCI30TEI) | √ | 109 | 106D | +1B4 _H | 0 to 15 (15) | √ | √ | | | |
| SCI31 | SCI31 receive error (INTSCI31ERI) | √ | 110 | 106E | +1B8 _H | 0 to 15 (15) | √ | √ | | | |
| | SCI31 receive data full (INTSCI31RXI) | | 111 | 106F | +1BC _H | 0 to 15 (15) | √ | √ | | | |
| | SCI31 transmit data empty (INTSCI31TXI) | | 112 | 1070 | +1C0 _H | 0 to 15 (15) | √ | √ | | | |
| | SCI31 transmit end (INTSCI31TEI) | √ | 113 | 1071 | +1C4 _H | 0 to 15 (15) | √ | √ | | | |
| RLIN30 | RLIN30 status interrupt (INTRLIN30UR2) | | 114 | 1072 | +1C8 _H | 0 to 15 (15) | √ | √ | | | |
| | RLIN30 receive completion interrupt (INTRLIN30UR1) | | 115 | 1073 | +1CC _H | 0 to 15 (15) | √ | √ | | | |
| | RLIN30 transmit interrupt (INTRLIN30UR0) | | 116 | 1074 | +1D0 _H | 0 to 15 (15) | √ | √ | | | |
| RLIN31 | RLIN31 status interrupt (INTRLIN31UR2) | | 117 | 1075 | +1D4 _H | 0 to 15 (15) | √ | √ | | | |
| | RLIN31 receive completion interrupt (INTRLIN31UR1) | | 118 | 1076 | +1D8 _H | 0 to 15 (15) | √ | √ | | | |
| | RLIN31 transmit interrupt (INTRLIN31UR0) | | 119 | 1077 | +1DC _H | 0 to 15 (15) | √ | √ | | | |
| SINT | Software interrupt 3 (INTSINT3) | √ | 120 | 1078 | +1E0 _H | 0 to 15 (15) | √ | √ | | | |
| | Software interrupt 4 (INTSINT4) | √ | 121 | 1079 | +1E4 _H | 0 to 15 (15) | √ | √ | | | |
| | Reserved | | 122 | 107A | +1E8 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 123 | 107B | +1EC _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 124 | 107C | +1F0 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 125 | 107D | +1F4 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 126 | 107E | +1F8 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 127 | 107F | +1FC _H | 0 to 15 (15) | — | — | | | |
| INTP | External interrupt 5 (INTP5) | | 128 | 1080 | +200 _H | 0 to 15 (15) | √ | √ | | | |
| | External interrupt 6 (INTP6) | | 129 | 1081 | +204 _H | 0 to 15 (15) | √ | √ | | | |
| | External interrupt 7 (INTP7) | | 130 | 1082 | +208 _H | 0 to 15 (15) | √ | √ | | | |
| | External interrupt 8 (INTP8) | | 131 | 1083 | +20C _H | 0 to 15 (15) | √ | √ | | | |
| | External interrupt 9 (INTP9) | | 132 | 1084 | +210 _H | 0 to 15 (15) | √ | √ | | | |
| TAUJ0 | CH0 interrupt of TAUJ0 (INTTAUJ0I0) | | 133 | 1085 | +214 _H | 0 to 15 (15) | √ | √ | | | |
| | CH1 interrupt of TAUJ0 (INTTAUJ0I1) | | 134 | 1086 | +218 _H | 0 to 15 (15) | √ | √ | | | |
| | CH2 interrupt of TAUJ0 (INTTAUJ0I2) | | 135 | 1087 | +21C _H | 0 to 15 (15) | √ | √ | | | |
| | CH3 interrupt of TAUJ0 (INTTAUJ0I3) | | 136 | 1088 | +220 _H | 0 to 15 (15) | √ | √ | | | |

Table 6.11 Interrupt Exception Handler and Priority (4/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | Table Reference | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|---------------------------------------|---|------------------------------|-------------------------------------|-------------------|---|---|-------------------|--|------------------|------|------|
| | | | | | Direct Branch | | | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| TAUJ1 | CH0 interrupt of TAUJ1 (INTTAUJ110) | | 137 | 1089 | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +224 _H | 0 to 15 (15) | High | √ | √ |
| | CH1 interrupt of TAUJ1 (INTTAUJ111) | | 138 | 108A | | | +228 _H | 0 to 15 (15) | | √ | √ |
| | CH2 interrupt of TAUJ1 (INTTAUJ112) | | 139 | 108B | | | +22C _H | 0 to 15 (15) | | √ | √ |
| | CH3 interrupt of TAUJ1 (INTTAUJ113) | | 140 | 108C | | | +230 _H | 0 to 15 (15) | | √ | √ |
| TAUD0 | CH0 interrupt of TAUD0 (INTTAUD010) | | 141 | 108D | | | +234 _H | 0 to 15 (15) | | √ | √ |
| | CH1 interrupt of TAUD0 (INTTAUD011) | | 142 | 108E | | | +238 _H | 0 to 15 (15) | | √ | √ |
| | CH2 interrupt of TAUD0 (INTTAUD012) | | 143 | 108F | | | +23C _H | 0 to 15 (15) | | √ | √ |
| | CH3 interrupt of TAUD0 (INTTAUD013) | | 144 | 1090 | | | +240 _H | 0 to 15 (15) | | √ | √ |
| | CH4 interrupt of TAUD0 (INTTAUD014) | | 145 | 1091 | | | +244 _H | 0 to 15 (15) | | √ | √ |
| | CH5 interrupt of TAUD0 (INTTAUD015) | | 146 | 1092 | | | +248 _H | 0 to 15 (15) | | √ | √ |
| | CH6 interrupt of TAUD0 (INTTAUD016) | | 147 | 1093 | | | +24C _H | 0 to 15 (15) | | √ | √ |
| | CH7 interrupt of TAUD0 (INTTAUD017) | | 148 | 1094 | | | +250 _H | 0 to 15 (15) | | √ | √ |
| | CH8 interrupt of TAUD0 (INTTAUD018) | | 149 | 1095 | | | +254 _H | 0 to 15 (15) | | √ | √ |
| | CH9 interrupt of TAUD0 (INTTAUD019) | | 150 | 1096 | | | +258 _H | 0 to 15 (15) | | √ | √ |
| | CH10 interrupt of TAUD0 (INTTAUD0110) | | 151 | 1097 | | | +25C _H | 0 to 15 (15) | | √ | √ |
| | CH11 interrupt of TAUD0 (INTTAUD0111) | | 152 | 1098 | +260 _H | 0 to 15 (15) | √ | √ | | | |
| | CH12 interrupt of TAUD0 (INTTAUD0112) | | 153 | 1099 | +264 _H | 0 to 15 (15) | √ | √ | | | |
| | CH13 interrupt of TAUD0 (INTTAUD0113) | | 154 | 109A | +268 _H | 0 to 15 (15) | √ | √ | | | |
| | CH14 interrupt of TAUD0 (INTTAUD0114) | | 155 | 109B | +26C _H | 0 to 15 (15) | √ | √ | | | |
| CH15 interrupt of TAUD0 (INTTAUD0115) | | 156 | 109C | +270 _H | 0 to 15 (15) | √ | √ | | | | |
| | Reserved | | 157 | 109D | +274 _H | 0 to 15 (15) | — | — | | | |
| TAUD1 | CH0 interrupt of TAUD1 (INTTAUD110) | | 158 | 109E | +278 _H | 0 to 15 (15) | √ | √ | | | |
| | CH1 interrupt of TAUD1 (INTTAUD111) | | 159 | 109F | +27C _H | 0 to 15 (15) | √ | √ | | | |
| | CH2 interrupt of TAUD1 (INTTAUD112) | | 160 | 10A0 | +280 _H | 0 to 15 (15) | √ | √ | | | |
| | CH3 interrupt of TAUD1 (INTTAUD113) | | 161 | 10A1 | +284 _H | 0 to 15 (15) | √ | √ | | | |
| | CH4 interrupt of TAUD1 (INTTAUD114) | | 162 | 10A2 | +288 _H | 0 to 15 (15) | √ | √ | | | |
| | CH5 interrupt of TAUD1 (INTTAUD115) | | 163 | 10A3 | +28C _H | 0 to 15 (15) | √ | √ | | | |
| | CH6 interrupt of TAUD1 (INTTAUD116) | | 164 | 10A4 | +290 _H | 0 to 15 (15) | √ | √ | | | |
| | CH7 interrupt of TAUD1 (INTTAUD117) | | 165 | 10A5 | +294 _H | 0 to 15 (15) | √ | √ | | | |
| | CH8 interrupt of TAUD1 (INTTAUD118) | | 166 | 10A6 | +298 _H | 0 to 15 (15) | √ | √ | | | |
| | CH9 interrupt of TAUD1 (INTTAUD119) | | 167 | 10A7 | +29C _H | 0 to 15 (15) | √ | √ | | | |
| | CH10 interrupt of TAUD1 (INTTAUD1110) | | 168 | 10A8 | +2A0 _H | 0 to 15 (15) | √ | √ | | | |
| | CH11 interrupt of TAUD1 (INTTAUD1111) | | 169 | 10A9 | +2A4 _H | 0 to 15 (15) | √ | √ | | | |
| | CH12 interrupt of TAUD1 (INTTAUD1112) | | 170 | 10AA | +2A8 _H | 0 to 15 (15) | √ | √ | | | |
| | CH13 interrupt of TAUD1 (INTTAUD1113) | | 171 | 10AB | +2AC _H | 0 to 15 (15) | √ | √ | | | |
| | CH14 interrupt of TAUD1 (INTTAUD1114) | | 172 | 10AC | +2B0 _H | 0 to 15 (15) | √ | √ | | | |
| CH15 interrupt of TAUD1 (INTTAUD1115) | | 173 | 10AD | +2B4 _H | 0 to 15 (15) | √ | √ | | | | |
| CSIG0 | CSIG0 communication error interrupt (INTCSIG0IRE) | | 174 | 10AE | +2B8 _H | 0 to 15 (15) | √ | √ | | | |
| | CSIG0 transmit status interrupt (INTCSIG0IC) | | 175 | 10AF | +2BC _H | 0 to 15 (15) | √ | √ | | | |
| | CSIG0 receive status interrupt (INTCSIG0IR) | | 176 | 10B0 | +2C0 _H | 0 to 15 (15) | √ | √ | | | |
| ADCG1 | ADCG1 error interrupt (INTADCG1ERR) | | 177 | 10B1 | +2C4 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG1 SG0 end interrupt (INTADCG110) | | 178 | 10B2 | +2C8 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG1 SG1 end interrupt (INTADCG111) | | 179 | 10B3 | +2CC _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG1 SG2 end interrupt (INTADCG112) | | 180 | 10B4 | +2D0 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG1 SG3 end interrupt (INTADCG113) | | 181 | 10B5 | +2D4 _H | 0 to 15 (15) | √ | √ | | | |
| | ADCG1 SG4 end interrupt (INTADCG114) | | 182 | 10B6 | +2D8 _H | 0 to 15 (15) | √ | √ | | | |
| RSCANFD | Channel error interrupt 0 (INTRCAN0ERR) | √ | 183 | 10B7 | +2DC _H | 0 to 15 (15) | √ | √ | | | |
| | COM RX FIFO interrupt 0 (INTRCAN0REC) | √ | 184 | 10B8 | +2E0 _H | 0 to 15 (15) | √ | √ | | | |
| | Channel TX interrupt 0 (INTRCAN0TRX) | √ | 185 | 10B9 | +2E4 _H | 0 to 15 (15) | √ | √ | | | |

Table 6.11 Interrupt Exception Handler and Priority (5/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|-------------------|--|------------------------------|-------------------------------------|-------------|---|---|-------------------|--|------------------|------|------|
| | | | | | Direct Branch | | Table Reference | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| RSCAN | Channel error interrupt 1 (INTRCAN1ERR) | √ | 186 | 10BA | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +2E8 _H | 0 to 15 (15) | High | √ | √ |
| | COM RX FIFO interrupt 1 (INTRCAN1REC) | √ | 187 | 10BB | | | +2EC _H | 0 to 15 (15) | | √ | √ |
| | Channel TX interrupt 1 (INTRCAN1TRX) | √ | 188 | 10BC | | | +2F0 _H | 0 to 15 (15) | | √ | √ |
| | Global error interrupt (INTRCANGERR) | √ | 189 | 10BD | | | +2F4 _H | 0 to 15 (15) | | √ | √ |
| | RX FIFO interrupt (INTRCANGRECC) | √ | 190 | 10BE | | | +2F8 _H | 0 to 15 (15) | | √ | √ |
| | Channel error interrupt 2 (INTRCAN2ERR) | √ | 191 | 10BF | | | +2FC _H | 0 to 15 (15) | | √ | √ |
| | COM RX FIFO interrupt 2 (INTRCAN2REC) | √ | 192 | 10C0 | | | +300 _H | 0 to 15 (15) | | √ | √ |
| | Channel TX interrupt 2 (INTRCAN2TRX) | √ | 193 | 10C1 | | | +304 _H | 0 to 15 (15) | | √ | √ |
| FLXA0 | FlexRay0 interrupt (INTFLXA0LINE0) | √ | 194 | 10C2 | +308 _H | 0 to 15 (15) | √ | √ | | | |
| | FlexRay1 interrupt (INTFLXA0LINE1) | √ | 195 | 10C3 | +30C _H | 0 to 15 (15) | √ | √ | | | |
| | Timer 0 interrupt (INTFLXA0TIM0) | √ | 196 | 10C4 | +310 _H | 0 to 15 (15) | √ | √ | | | |
| | Timer 1 interrupt (INTFLXA0TIM1) | √ | 197 | 10C5 | +314 _H | 0 to 15 (15) | √ | √ | | | |
| | Timer 2 interrupt (INTFLXA0TIM2) | √ | 198 | 10C6 | +318 _H | 0 to 15 (15) | √ | √ | | | |
| | FIFO transfer interrupt (INTFLXA0FDA) | √ | 199 | 10C7 | +31C _H | 0 to 15 (15) | √ | √ | | | |
| | FIFO transfer warning interrupt (INTFLXA0FW) | √ | 200 | 10C8 | +320 _H | 0 to 15 (15) | √ | √ | | | |
| | Output transfer warning interrupt (INTFLXA0OW) | √ | 201 | 10C9 | +324 _H | 0 to 15 (15) | √ | √ | | | |
| | Output transfer end interrupt (INTFLXA0OT) | √ | 202 | 10CA | +328 _H | 0 to 15 (15) | √ | √ | | | |
| | Input queue full interrupt (INTFLXA0IQF) | √ | 203 | 10CB | +32C _H | 0 to 15 (15) | √ | √ | | | |
| | Input queue empty interrupt (INTFLXA0IQE) | √ | 204 | 10CC | +330 _H | 0 to 15 (15) | √ | √ | | | |
| | Reserved | | | 205 | 10CD | +334 _H | 0 to 15 (15) | — | — | | |
| | Reserved | | | 206 | 10CE | +338 _H | 0 to 15 (15) | — | — | | |
| | Reserved | | | 207 | 10CF | +33C _H | 0 to 15 (15) | — | — | | |
| | Reserved | | | 208 | 10D0 | +340 _H | 0 to 15 (15) | — | — | | |
| | Reserved | | | 209 | 10D1 | +344 _H | 0 to 15 (15) | — | — | | |
| Reserved | | | 210 | 10D2 | +348 _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 211 | 10D3 | +34C _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 212 | 10D4 | +350 _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 213 | 10D5 | +354 _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 214 | 10D6 | +358 _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 215 | 10D7 | +35C _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 216 | 10D8 | +360 _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 217 | 10D9 | +364 _H | 0 to 15 (15) | — | — | | | |
| SCI32 | SCI32 receive error (INTSCI32ERI) | √ | 218 | 10DA | +368 _H | 0 to 15 (15) | √ | √ | | | |
| | SCI32 receive data full (INTSCI32RXI) | | 219 | 10DB | +36C _H | 0 to 15 (15) | √ | √ | | | |
| | SCI32 transmit data empty (INTSCI32TXI) | | 220 | 10DC | +370 _H | 0 to 15 (15) | √ | √ | | | |
| | SCI32 transmit end (INTSCI32TEI) | √ | 221 | 10DD | +374 _H | 0 to 15 (15) | √ | √ | | | |
| | Reserved | | | 222 | 10DE | +378 _H | 0 to 15 (15) | — | — | | |
| Reserved | | | 223 | 10DF | +37C _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 224 | 10E0 | +380 _H | 0 to 15 (15) | — | — | | | |
| Reserved | | | 225 | 10E1 | +384 _H | 0 to 15 (15) | — | — | | | |
| PSI50 | PSI50 status interrupt (INTPSI50SI) | √ | 226 | 10E2 | +388 _H | 0 to 15 (15) | √ | √ | | | |
| | PSI50 receive interrupt (INTPSI50RI) | √ | 227 | 10E3 | +38C _H | 0 to 15 (15) | √ | √ | | | |
| | PSI50 transfer interrupt (INTPSI50TI) | √ | 228 | 10E4 | +390 _H | 0 to 15 (15) | √ | √ | | | |
| PSI51 | PSI51 status interrupt (INTPSI51SI) | √ | 229 | 10E5 | +394 _H | 0 to 15 (15) | √ | √ | | | |
| | PSI51 receive interrupt (INTPSI51RI) | √ | 230 | 10E6 | +398 _H | 0 to 15 (15) | √ | √ | | | |
| | PSI51 transfer interrupt (INTPSI51TI) | √ | 231 | 10E7 | +39C _H | 0 to 15 (15) | √ | √ | | | |
| SENT0 | SETN0 status interrupt (INTSENT0SI) | √ | 232 | 10E8 | +3A0 _H | 0 to 15 (15) | √ | √ | | | |
| | SETN0 receive interrupt (INTSENT0RI) | | 233 | 10E9 | +3A4 _H | 0 to 15 (15) | √ | √ | | | |

Table 6.11 Interrupt Exception Handler and Priority (6/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB | |
|---------------------------------------|---|------------------------------|-------------------------------------|-------------------|---|---|-------------------|--|------------------|--------------|------|---|
| | | | | | Direct Branch | | Table Reference | | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | | |
| SENT1 | SETN1 status interrupt (INTSENT1SI) | √ | 234 | 10EA | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +3A8 _H | 0 to 15 (15) | High | √ | √ | |
| | SETN1 receive interrupt (INTSENT1RI) | | 235 | 10EB | | | | +3AC _H | | 0 to 15 (15) | √ | √ |
| SENT2 | SETN2 status interrupt (INTSENT2SI) | √ | 236 | 10EC | | | | +3B0 _H | | 0 to 15 (15) | √ | √ |
| | SETN2 receive interrupt (INTSENT2RI) | | 237 | 10ED | | | | +3B4 _H | | 0 to 15 (15) | √ | √ |
| SENT3 | SETN3 status interrupt (INTSENT3SI) | √ | 238 | 10EE | | | | +3B8 _H | | 0 to 15 (15) | √ | √ |
| | SETN3 receive interrupt (INTSENT3RI) | | 239 | 10EF | | | | +3BC _H | | 0 to 15 (15) | √ | √ |
| SENT4 | SETN4 status interrupt (INTSENT4SI) | √ | 240 | 10F0 | | | | +3C0 _H | | 0 to 15 (15) | √ | √ |
| | SETN4 receive interrupt (INTSENT4RI) | | 241 | 10F1 | | | | +3C4 _H | | 0 to 15 (15) | √ | √ |
| SENT5 | SETN5 status interrupt (INTSENT5SI) | √ | 242 | 10F2 | | | | +3C8 _H | | 0 to 15 (15) | √ | √ |
| | SETN5 receive interrupt (INTSENT5RI) | | 243 | 10F3 | | | | +3CC _H | | 0 to 15 (15) | √ | √ |
| DTS | DTS transfer end interrupt Ch0-31 (INTDTSTC0) | √ | 244 | 10F4 | | | | +3D0 _H | | 0 to 15 (15) | √ | √ |
| | DTS transfer end interrupt Ch32-63 (INTDTSTC1) | √ | 245 | 10F5 | | | | +3D4 _H | | 0 to 15 (15) | √ | √ |
| | DTS transfer end interrupt Ch64-95 (INTDTSTC2) | √ | 246 | 10F6 | | | | +3D8 _H | | 0 to 15 (15) | √ | √ |
| | DTS transfer end interrupt Ch96-127 (INTDTSTC3) | √ | 247 | 10F7 | | | | +3DC _H | | 0 to 15 (15) | √ | √ |
| | DTS count match interrupt Ch0-31 (INTDTSCM0) | √ | 248 | 10F8 | | | | +3E0 _H | | 0 to 15 (15) | √ | √ |
| | DTS count match interrupt Ch32-63 (INTDTSCM1) | √ | 249 | 10F9 | | | | +3E4 _H | | 0 to 15 (15) | √ | √ |
| | DTS count match interrupt Ch64-95 (INTDTSCM2) | √ | 250 | 10FA | | | | +3E8 _H | | 0 to 15 (15) | √ | √ |
| | DTS count match interrupt Ch96-127 (INTDTSCM3) | √ | 251 | 10FB | | | | +3EC _H | | 0 to 15 (15) | √ | √ |
| SINT | Software interrupt 0 (INTSINT0) | √ | 252 | 10FC | | | | +3F0 _H | | 0 to 15 (15) | √ | √ |
| | Software interrupt 1 (INTSINT1) | √ | 253 | 10FD | | | | +3F4 _H | | 0 to 15 (15) | √ | √ |
| | Software interrupt 2 (INTSINT2) | √ | 254 | 10FE | | | | +3F8 _H | | 0 to 15 (15) | √ | √ |
| | Reserved | | 255 | 10FF | | | | +3FC _H | | 0 to 15 (15) | — | — |
| TAUJ2 | CH0 interrupt of TAUJ2 (INTTAUJ2I0) | | 256 | 1100 | | | | +400 _H | | 0 to 15 (15) | √ | √ |
| | CH1 interrupt of TAUJ2 (INTTAUJ2I1) | | 257 | 1101 | | | | +404 _H | | 0 to 15 (15) | √ | √ |
| | CH2 interrupt of TAUJ2 (INTTAUJ2I2) | | 258 | 1102 | | | | +408 _H | | 0 to 15 (15) | √ | √ |
| | CH3 interrupt of TAUJ2 (INTTAUJ2I3) | | 259 | 1103 | | | | +40C _H | | 0 to 15 (15) | √ | √ |
| TAUD2 | CH0 interrupt of TAUD2 (INTTAUD2I0) | | 260 | 1104 | | | | +410 _H | | 0 to 15 (15) | √ | √ |
| | CH1 interrupt of TAUD2 (INTTAUD2I1) | | 261 | 1105 | | | | +414 _H | | 0 to 15 (15) | √ | √ |
| | CH2 interrupt of TAUD2 (INTTAUD2I2) | | 262 | 1106 | +418 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH3 interrupt of TAUD2 (INTTAUD2I3) | | 263 | 1107 | +41C _H | 0 to 15 (15) | √ | √ | | | | |
| | CH4 interrupt of TAUD2 (INTTAUD2I4) | | 264 | 1108 | +420 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH5 interrupt of TAUD2 (INTTAUD2I5) | | 265 | 1109 | +424 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH6 interrupt of TAUD2 (INTTAUD2I6) | | 266 | 110A | +428 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH7 interrupt of TAUD2 (INTTAUD2I7) | | 267 | 110B | +42C _H | 0 to 15 (15) | √ | √ | | | | |
| | CH8 interrupt of TAUD2 (INTTAUD2I8) | | 268 | 110C | +430 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH9 interrupt of TAUD2 (INTTAUD2I9) | | 269 | 110D | +434 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH10 interrupt of TAUD2 (INTTAUD2I10) | | 270 | 110E | +438 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH11 interrupt of TAUD2 (INTTAUD2I11) | | 271 | 110F | +43C _H | 0 to 15 (15) | √ | √ | | | | |
| | CH12 interrupt of TAUD2 (INTTAUD2I12) | | 272 | 1110 | +440 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH13 interrupt of TAUD2 (INTTAUD2I13) | | 273 | 1111 | +444 _H | 0 to 15 (15) | √ | √ | | | | |
| | CH14 interrupt of TAUD2 (INTTAUD2I14) | | 274 | 1112 | +448 _H | 0 to 15 (15) | √ | √ | | | | |
| CH15 interrupt of TAUD2 (INTTAUD2I15) | | 275 | 1113 | +44C _H | 0 to 15 (15) | √ | √ | | | | | |
| ENCA0 | ENCA0 overflow interrupt (INTENCA0IOV) | | 276 | 1114 | +450 _H | 0 to 15 (15) | √ | √ | | | | |
| | ENCA0 match/capture interrupt 0 (INTENCA0I0) | | 277 | 1115 | +454 _H | 0 to 15 (15) | √ | √ | | | | |
| | ENCA0 match/capture interrupt 1 (INTENCA0I1) | | 278 | 1116 | +458 _H | 0 to 15 (15) | √ | √ | | | | |
| | ENCA0 underflow interrupt (INTENCA0IUD) | | 279 | 1117 | +45C _H | 0 to 15 (15) | √ | √ | | | | |
| | ENCA0 encoder clear interrupt (INTENCA0IEC) | | 280 | 1118 | +460 _H | 0 to 15 (15) | √ | √ | | | | |

Table 6.11 Interrupt Exception Handler and Priority (7/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|-------------------|--|------------------------------|-------------------------------------|-------------------|---|---|-------------------|--|------------------|------|------|
| | | | | | Direct Branch | | Table Reference | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| ENCA1 | ENCA1 overflow interrupt (INTENCA1IOV) | | 281 | 1119 | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +464 _H | 0 to 15 (15) | High | √ | √ |
| | ENCA1 match/capture interrupt 0 (INTENCA1I0) | | 282 | 111A | | | +468 _H | 0 to 15 (15) | | √ | √ |
| | ENCA1 match/capture interrupt 1 (INTENCA1I1) | | 283 | 111B | | | +46C _H | 0 to 15 (15) | | √ | √ |
| | ENCA1 underflow interrupt (INTENCA1IUD) | | 284 | 111C | | | +470 _H | 0 to 15 (15) | | √ | √ |
| | ENCA1 encoder clear interrupt (INTENCA1IEC) | | 285 | 111D | | | +474 _H | 0 to 15 (15) | | √ | √ |
| TPBA0 | Period match detection interrupt 0 (INTTPBA0IPRD) | | 286 | 111E | | | +478 _H | 0 to 15 (15) | | √ | √ |
| | Duty match detection interrupt 0 (INTTPBA0IDTY) | | 287 | 111F | | | +47C _H | 0 to 15 (15) | | √ | √ |
| | Pattern number matching detection interrupt 0 (INTTPBA0IPAT) | | 288 | 1120 | | | +480 _H | 0 to 15 (15) | | √ | √ |
| TPBA1 | Period match detection interrupt 1 (INTTPBA1IPRD) | | 289 | 1121 | | | +484 _H | 0 to 15 (15) | | √ | √ |
| | Duty match detection interrupt 1 (INTTPBA1IDTY) | | 290 | 1122 | | | +488 _H | 0 to 15 (15) | | √ | √ |
| | Pattern number matching detection interrupt 1 (INTTPBA1IPAT) | | 291 | 1123 | | | +48C _H | 0 to 15 (15) | | √ | √ |
| | Reserved | | 292 | 1124 | | | +490 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 293 | 1125 | | | +494 _H | 0 to 15 (15) | | — | — |
| MPX | ADCG0 MPX request interrupt (INTADCG0MPX) | | 294 | 1126 | | | +498 _H | 0 to 15 (15) | | √ | √ |
| | ADCG1 MPX request interrupt (INTADCG1MPX) | | 295 | 1127 | | | +49C _H | 0 to 15 (15) | | √ | √ |
| INTP | External interrupt 10 (INTP10) | | 296 | 1128 | | | +4A0 _H | 0 to 15 (15) | | √ | √ |
| | External interrupt 11 (INTP11) | | 297 | 1129 | | | +4A4 _H | 0 to 15 (15) | | √ | √ |
| | External interrupt 12 (INTP12) | | 298 | 112A | | | +4A8 _H | 0 to 15 (15) | | √ | √ |
| | Reserved | | 299 | 112B | | | +4AC _H | 0 to 15 (15) | | — | — |
| | Reserved | | 300 | 112C | | | +4B0 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 301 | 112D | | | +4B4 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 302 | 112E | | | +4B8 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 303 | 112F | | | +4BC _H | 0 to 15 (15) | | — | — |
| | Reserved | | 304 | 1130 | | | +4C0 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 305 | 1131 | | | +4C4 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 306 | 1132 | | | +4C8 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 307 | 1133 | | | +4CC _H | 0 to 15 (15) | | — | — |
| | Reserved | | 308 | 1134 | | | +4D0 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 309 | 1135 | +4D4 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 310 | 1136 | +4D8 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 311 | 1137 | +4DC _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 312 | 1138 | +4E0 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 313 | 1139 | +4E4 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 314 | 113A | +4E8 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 315 | 113B | +4EC _H | 0 to 15 (15) | — | — | | | |
| Reserved | | 316 | 113C | +4F0 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 317 | 113D | +4F4 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 318 | 113E | +4F8 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 319 | 113F | +4FC _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 320 | 1140 | +500 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 321 | 1141 | +504 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 322 | 1142 | +508 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 323 | 1143 | +50C _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 324 | 1144 | +510 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 325 | 1145 | +514 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 326 | 1146 | +518 _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 327 | 1147 | +51C _H | 0 to 15 (15) | — | — | | | | |
| Reserved | | 328 | 1148 | +520 _H | 0 to 15 (15) | — | — | | | | |

Table 6.11 Interrupt Exception Handler and Priority (8/9)

| Functional Module | Interrupt Source Name | Level Interrupt ¹ | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|-------------------|-----------------------|------------------------------|-------------------------------------|-------------|---|---|-------------------|--|------------------|------|------|
| | | | | | Direct Branch | | Table Reference | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| | Reserved | | 329 | 1149 | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +524 _H | 0 to 15 (15) | High | — | — |
| | Reserved | | 330 | 114A | | | +528 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 331 | 114B | | | +52C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 332 | 114C | | | +530 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 333 | 114D | | | +534 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 334 | 114E | | | +538 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 335 | 114F | | | +53C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 336 | 1150 | | | +540 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 337 | 1151 | | | +544 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 338 | 1152 | | | +548 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 339 | 1153 | | | +54C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 340 | 1154 | | | +550 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 341 | 1155 | | | +554 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 342 | 1156 | | | +558 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 343 | 1157 | | | +55C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 344 | 1158 | | | +560 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 345 | 1159 | | | +564 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 346 | 115A | | | +568 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 347 | 115B | | | +56C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 348 | 115C | | | +570 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 349 | 115D | | | +574 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 350 | 115E | | | +578 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 351 | 115F | | | +57C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 352 | 1160 | | | +580 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 353 | 1161 | | | +584 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 354 | 1162 | | | +588 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 355 | 1163 | | | +58C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 356 | 1164 | | | +590 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 357 | 1165 | | | +594 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 358 | 1166 | | | +598 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 359 | 1167 | | | +59C _H | 0 to 15 (15) | — | — | |
| | Reserved | | 360 | 1168 | | | +5A0 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 361 | 1169 | | | +5A4 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 362 | 116A | | | +5A8 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 363 | 116B | | | +5AC _H | 0 to 15 (15) | — | — | |
| | Reserved | | 364 | 116C | | | +5B0 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 365 | 116D | | | +5B4 _H | 0 to 15 (15) | — | — | |
| | Reserved | | 366 | 116E | +5B8 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 367 | 116F | +5BC _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 368 | 1170 | +5C0 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 369 | 1171 | +5C4 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 370 | 1172 | +5C8 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 371 | 1173 | +5CC _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 372 | 1174 | +5D0 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 373 | 1175 | +5D4 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 374 | 1176 | +5D8 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 375 | 1177 | +5DC _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 376 | 1178 | +5E0 _H | 0 to 15 (15) | — | — | | | |
| | Reserved | | 377 | 1179 | +5E4 _H | 0 to 15 (15) | — | — | | | |

Table 6.11 Interrupt Exception Handler and Priority (9/9)

| Functional Module | Interrupt Source Name | Level Interrupt ^{*1} | Number of Interrupt Channel (EIINT) | Source Code | Offset Address | | | Interrupt Priority (Value after reset) | Default Priority | 1 MB | 2 MB |
|-------------------|--|-------------------------------|-------------------------------------|-------------|---|---|-------------------|--|------------------|------|------|
| | | | | | Direct Branch | | Table Reference | | | | |
| | | | | | RINT = 0 | RINT = 1 | | | | | |
| | Reserved | | 378 | 117A | The offset address is the same in all channels and is determined between +100 _H and +1F0 _H according to priority. | The offset address is always +100 _H regardless of priority to reduce the offset address. | +5E8 _H | 0 to 15 (15) | High | — | — |
| Flash | Flash sequencer end interrupt (INTFLENDNM) ^{*3} | | 379 | 117B | | | +5EC _H | 0 to 15 (15) | | √ | √ |
| | Reserved | | 380 | 117C | | | +5F0 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 381 | 117D | | | +5F4 _H | 0 to 15 (15) | | — | — |
| | Reserved | | 382 | 117E | | | +5F8 _H | 0 to 15 (15) | | — | — |
| Flash | Flash sequencer end error interrupt (INTFLERR) ^{*3} | √ | 383 | 117F | | | +5FC _H | 0 to 15 (15) | | √ | √ |

Note 1. The table indicates whether interrupts from each source are detected by level or edge. If a cell in this column is checked, the interrupt from that source is detected when the signal is at the high level. If it is blank, detection is of the selected edge. For interrupts from sources detected by the signal being at the high level, use software in the interrupt handler to clear the status register in the given module.

Note 2. For a level interrupt, the status register in each module should be cleared during interrupt processing by software.

Note 3. For details of interrupt generation factors, see *the RH850/P1M-E Flash Memory User's Manual: Hardware Interface*.

6.5 Operation

6.5.1 External Interrupts (NMI / INTP)

The external interrupts INTPm and NMI can be configured to generate an interrupt request upon a rising or falling edge or upon both edges of the external pin.

For the interrupt detection flow, see **Section 6.5.4, Interrupt Processing Flow**.

6.5.2 Software Interrupt

For the operation of software interrupts, see **Section 6.2.7, SINTR0 - SINTR4 — Software Interrupt Registers** and **Section 6.5.4, Interrupt Processing Flow**.

6.5.3 DTS Interrupt Merge Function

Up to 128 transfer end interrupts and up to 128 transfer count match interrupts are aggregated into one interrupt in units of 32 interrupts.

When multiplex interrupts have occurred, bits of only accepted interrupt sources are set to 1 in status registers (PINT0 to PINT7) so that accepted interrupts can be recognized. For the interrupt request flow by merging DTS interrupts, see **Section 6.5.4, Interrupt Processing Flow**.

Table 6.12 DTS Interrupt-related Registers

| Interrupt Source | CH | Status Register | Clear Register |
|------------------------------------|-----------|-----------------|----------------|
| DTS transfer end interrupt | 0 to 31 | PINT0 | PINTCLR0 |
| | 32 to 63 | PINT1 | PINTCLR1 |
| | 64 to 95 | PINT2 | PINTCLR2 |
| | 96 to 127 | PINT3 | PINTCLR3 |
| DTS transfer count match interrupt | 0 to 31 | PINT4 | PINTCLR4 |
| | 32 to 63 | PINT5 | PINTCLR5 |
| | 64 to 95 | PINT6 | PINTCLR6 |
| | 96 to 127 | PINT7 | PINTCLR7 |

6.5.4 Interrupt Processing Flow

6.5.4.1 NMI Processing Flow

Figure 6.1 shows an NMI detection flow.

- Select an NMI detecting method (rising edge, falling edge, and both edges) by setting the FCLA0CTL0 register.
- After an NMI has been detected, an interrupt request is sent to the INTC.
- When the interrupt processing in the INTC has been completed, the INTC enters the NMI detection waiting state again.

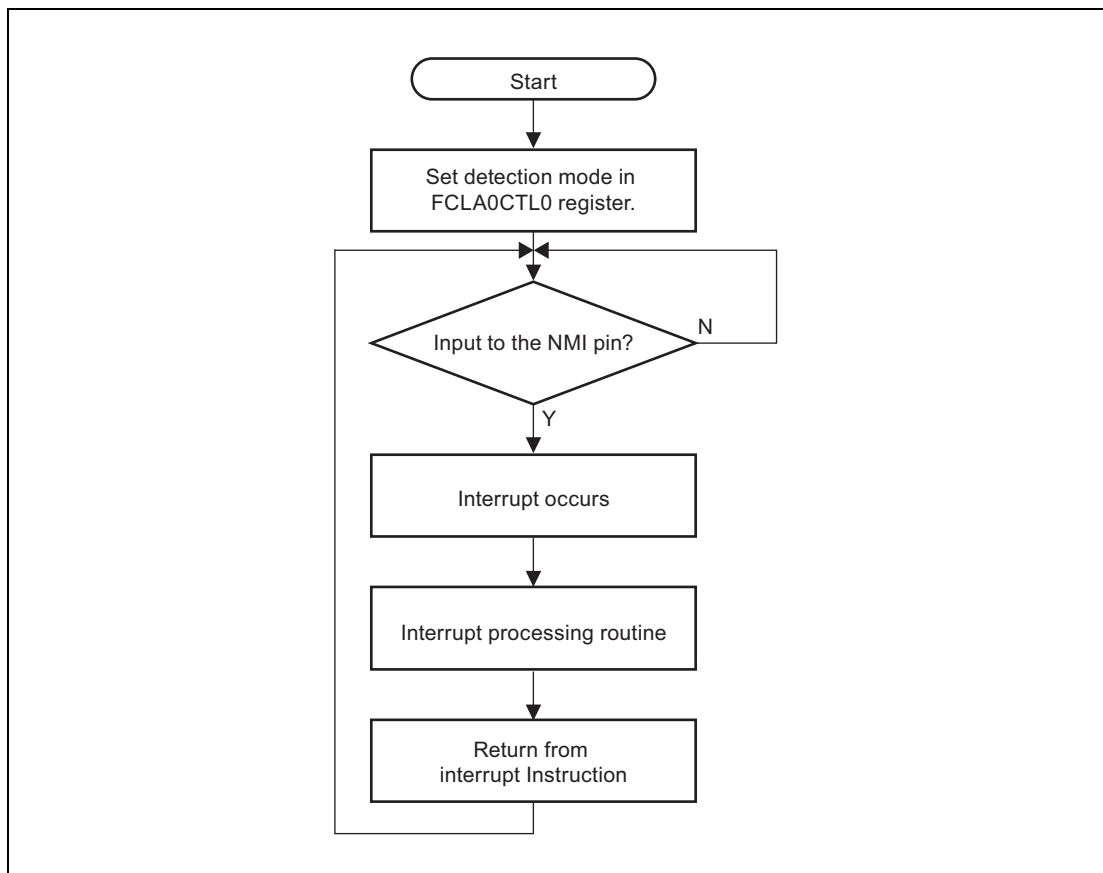


Figure 6.1 NMI Processing Flow

6.5.4.2 External Interrupt Processing Flow

Figure 6.2 shows an INTP (external interrupt) detection flow.

- Select an INTP detecting method (rising edge, falling edge, and both edges) by setting the FCLAnCTLm register.
- After an INTP has been detected, an interrupt request is sent to the INTC.

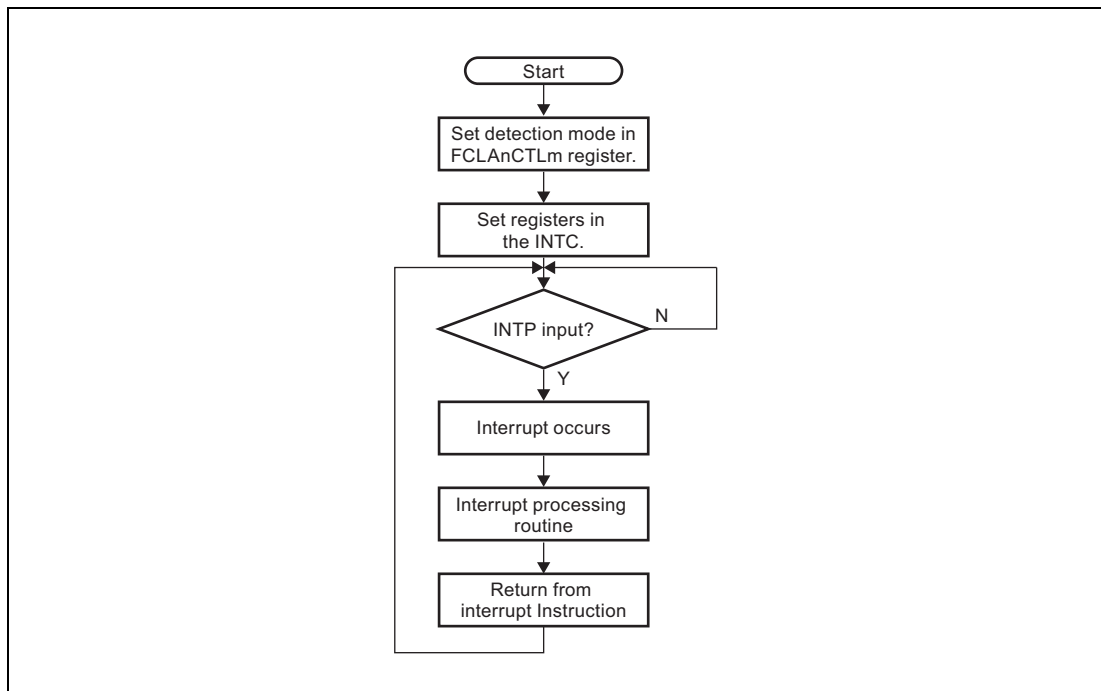


Figure 6.2 External Interrupt Processing Flow

6.5.4.3 Software Interrupt Processing Flow

Figure 6.3 shows the flow of software interrupt requests.

- Software interrupt requests are controlled by writing 00_H or 01_H to the counter bits SINTCn[7:0] in the software interrupt registers (SINTR0 to SINTR4).
- Writing 01_H increments the counter value by 1.
- Writing 00_H decrements the counter value by 1.
- If an incremented counter value is larger than 1, an interrupt request is issued to the INTC.
- On completion of other interrupt processing in the INTC, if SINTRn is 00_H, normal processing resumes.

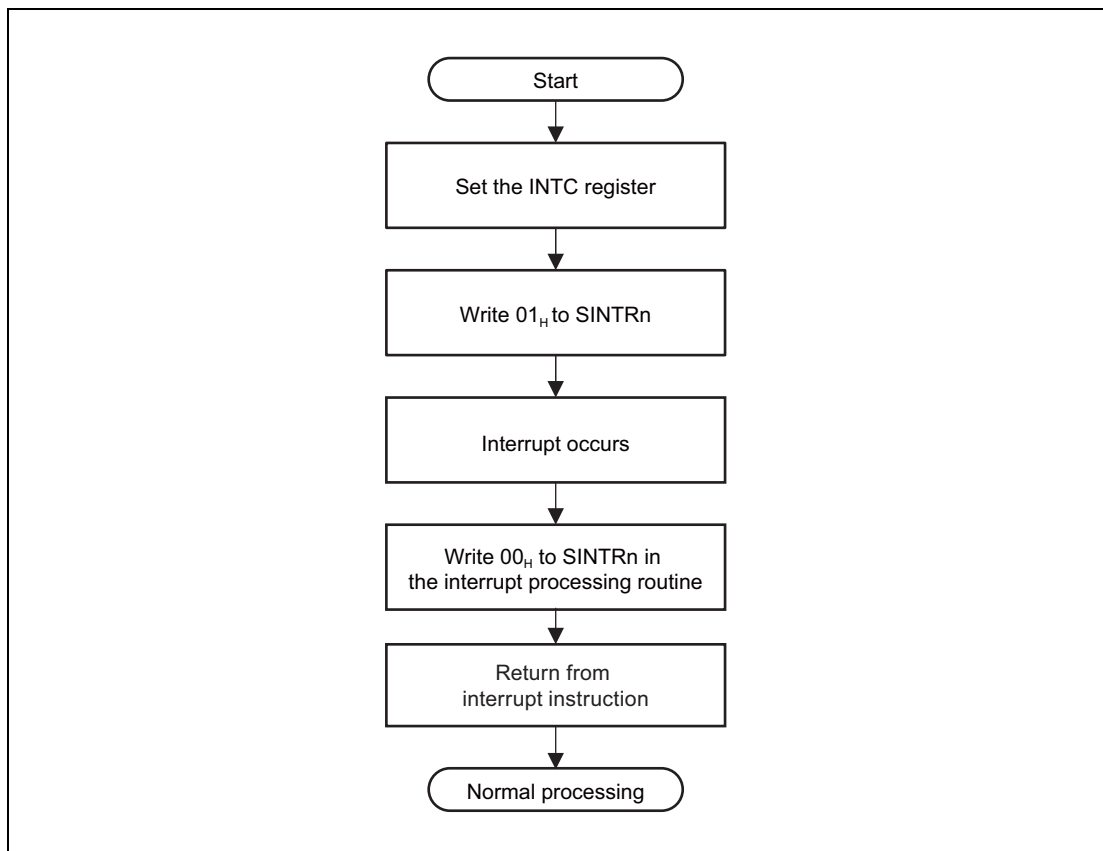


Figure 6.3 Software Interrupt Processing Flow

6.5.4.4 DTS Interrupt Processing Flow

Figure 6.4 shows a DTS interrupt request flow.

- When only one interrupt request is generated out of bundled 32 interrupt sources
 - The bit corresponding to the interrupt request in the PINTn register is set to 1 and an interrupt request is output.
 - On completion of other interrupt processing, write 1 to the interrupt clear register (PINTCLRn) to clear the interrupt request before issuing the return from interrupt instruction, then wait for the next one.
- When multiplex interrupt requests are generated out of bundled 32 interrupt sources
 - The highest-priority bit (interrupts on the lower-bits side take precedence) out of bits with interrupt request is extracted and only the extracted bit in the PINTn register is set to 1, and an interrupt request is output.
 - On completion of other interrupt processing, write 1 to the interrupt clear register (PINTCLRn) to clear the highest-priority interrupt request, then issue the return from interrupt instruction.
 - After the highest-priority interrupt request has been cleared, the second highest-priority interrupt request is accepted. The corresponding bit in the PINTn register is set to 1 in the same way as before, and an interrupt request is output.
 - These steps are repeated until all interrupt sources bundled into 32 bits are cleared.

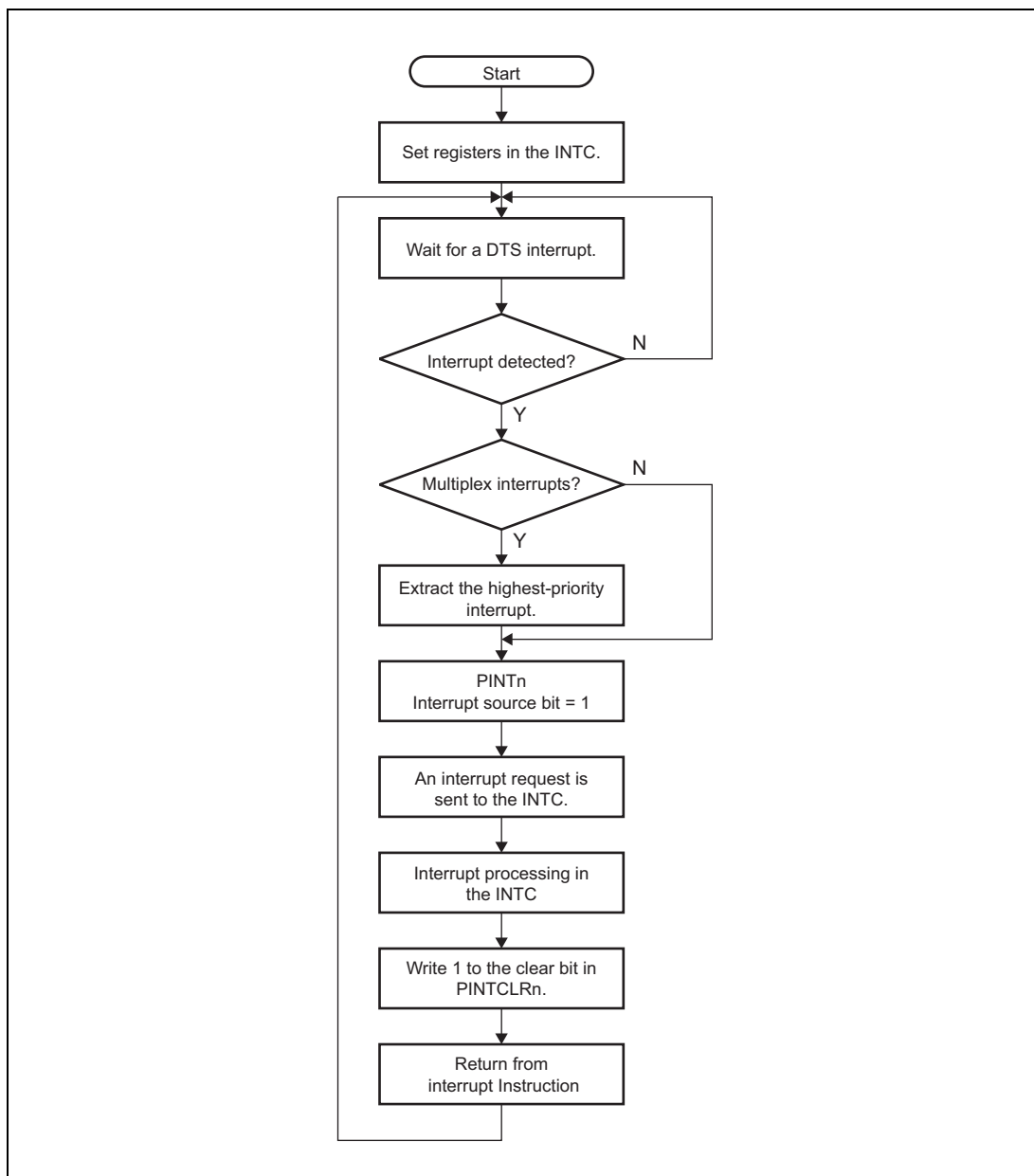


Figure 6.4 DTS Interrupt Processing Flow

6.6 Interrupt latency

The interrupt latency is summarized in **Table 6.13, Interrupt Latency**. The given values describe the response time from the assertion of the interrupt trigger at the INTC module and the instruction fetch of the first instruction inside the interrupt handling routine. The response time is the typical time for the given scenario. However, it is assumed that no other bus master accesses the code flash in case of an instruction cache miss. The unit of the response time is CPU1 clock cycle.

Table 6.13 Interrupt Latency

| Instruction cache | Interrupt detection | Exception handler addressing method | Address reference table location | Latency (CPU1) | |
|-------------------|---------------------|-------------------------------------|----------------------------------|----------------|-------|
| | | | | INTC1 | INTC2 |
| hit | edge | direct vector | NA | 7 | 12 |
| miss | edge | direct vector | NA | 10 | 15 |
| hit | level | direct vector | NA | 6 | 10 |
| miss | level | direct vector | NA | 9 | 13 |
| hit | edge | table reference | Local RAM | 11 | 16 |
| miss | edge | table reference | Local RAM | 14 | 19 |
| hit | level | table reference | Local RAM | 10 | 14 |
| miss | level | table reference | Local RAM | 13 | 17 |
| hit | edge | table reference | Code flash | 14 | 19 |
| miss | edge | table reference | Code flash | 17 | 22 |
| hit | level | table reference | Code flash | 13 | 17 |
| miss | level | table reference | Code flash | 16 | 20 |

6.7 Using Interrupt Request Signals to Initiate Data Transfer

An interrupt request signal activates the DMAC/DTS to perform data transfer. For details, see **Section 7, DMA**.

Section 7 DMA

7.1 Features of RH850/P1M-E DMA

7.1.1 Number of Channels

The products of the RH850/P1M-E series incorporate a DMA with the number of channels indicated below.

Table 7.1 Number of Channels

| Product Name | | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|--------------------|-----|-------------------------|-------------------------|
| Number of channels | DMA | 16ch Redundant | |
| | DTS | 128ch Redundant | |

Table 7.2 Indices

| Index | Description |
|-------|--|
| n | In this section, the DMA unit are identified with an index "n" (n = 0 to 15), for example, as DMACn.* ¹ |
| m | In this section, the DTS channels are identified with an index "m" (m = 000 to 127). |
| k | In this section, the DMA hardware transfer resource are identified with an index "k" (k = 0 to 127). |

Note 1. The correspondence between the channel number "n" and the channel is as follows.

| Channel Number n | Channel |
|------------------|-----------------|
| 0 | DMAC0 channel 0 |
| 1 | DMAC0 channel 1 |
| 2 | DMAC0 channel 2 |
| 3 | DMAC0 channel 3 |
| 4 | DMAC0 channel 4 |
| 5 | DMAC0 channel 5 |
| 6 | DMAC0 channel 6 |
| 7 | DMAC0 channel 7 |
| 8 | DMAC1 channel 0 |
| 9 | DMAC1 channel 1 |
| 10 | DMAC1 channel 2 |
| 11 | DMAC1 channel 3 |
| 12 | DMAC1 channel 4 |
| 13 | DMAC1 channel 5 |
| 14 | DMAC1 channel 6 |
| 15 | DMAC1 channel 7 |

7.1.2 Register Base Address

The DMA/DTS base address is indicated in the following table.

The DMA/DTS register address is expressed as an offset of the base address.

Table 7.3 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <DMAC_base> | FFFF 8000 _H |
| <DTS_base> | FFFF 9000 _H |
| <DMATRG_base> | FFF9 9000 _H |

7.1.3 Interrupt Requests

The DMA/DTS interrupt requests are listed in the following table.

Table 7.4 Interrupt Requests (1/2)

| Unit Interrupt Name | Description | Interrupt Number | ECM* ¹ |
|---------------------|--|------------------|-------------------|
| INTDMA0 | DMA00 transfer end/count match | 16 | — |
| INTDMA1 | DMA01 transfer end/count match | 17 | — |
| INTDMA2 | DMA02 transfer end/count match | 18 | — |
| INTDMA3 | DMA03 transfer end/count match | 19 | — |
| INTDMA4 | DMA04 transfer end/count match | 20 | — |
| INTDMA5 | DMA05 transfer end/count match | 21 | — |
| INTDMA6 | DMA06 transfer end/count match | 22 | — |
| INTDMA7 | DMA07 transfer end/count match | 23 | — |
| INTDMA8 | DMA08 transfer end/count match | 24 | — |
| INTDMA9 | DMA09 transfer end/count match | 25 | — |
| INTDMA10 | DMA10 transfer end/count match | 26 | — |
| INTDMA11 | DMA11 transfer end/count match | 27 | — |
| INTDMA12 | DMA12 transfer end/count match | 28 | — |
| INTDMA13 | DMA13 transfer end/count match | 29 | — |
| INTDMA14 | DMA14 transfer end/count match | 30 | — |
| INTDMA15 | DMA15 transfer end/count match | 31 | — |
| INTDTSTC0 | DTS transfer end interrupt CH0-CH31 | 244 | — |
| INTDTSTC1 | DTS transfer end interrupt CH32-CH63 | 245 | — |
| INTDTSTC2 | DTS transfer end interrupt CH64-CH95 | 246 | — |
| INTDTSTC3 | DTS transfer end interrupt CH96-CH127 | 247 | — |
| INTDTSCM0 | DTS count match interrupt CH0-CH31 | 248 | — |
| INTDTSCM1 | DTS count match interrupt CH32-CH63 | 249 | — |
| INTDTSCM2 | DTS count match interrupt CH64-CH95 | 250 | — |

Table 7.4 Interrupt Requests (2/2)

| Unit Interrupt Name | Description | Interrupt Number | ECM* ¹ |
|---------------------|---|------------------|-------------------|
| INTDTSCM3 | DTS count match interrupt CH96-CH127 | 251 | — |
| DMAERR | DMA transfer error | — | ○ |
| DMAVIOL | DMA illegal access error | — | ○ |

Note 1. For detail of the error input to ECM, see **Section 32, Error Control Module (ECM)**.

7.1.4 DMA Trigger Source

The table below shows the assignment of trigger sources to DMAC inputs.

The DMA trigger source assignment for each DMAC channel is set in the DTFR register.

Table 7.5 List of DMA Trigger Sources (1/4)

| Channel | Interrupt Name | DMA Trigger Source |
|-------------|----------------|---|
| DMACTRG[0] | INTP9 | External interrupt |
| DMACTRG[1] | INTTAUJ010 | TAUJ0 CH0 interrupt |
| DMACTRG[2] | INTTAUJ011 | TAUJ0 CH1 interrupt |
| DMACTRG[3] | INTTAUJ012 | TAUJ0 CH2 interrupt |
| DMACTRG[4] | INTTAUJ013 | TAUJ0 CH3 interrupt |
| DMACTRG[5] | INTTAUJ110 | TAUJ1 CH0 interrupt |
| DMACTRG[6] | INTTAUJ111 | TAUJ1 CH1 interrupt |
| DMACTRG[7] | INTTAUJ112 | TAUJ1 CH2 interrupt |
| DMACTRG[8] | INTTAUJ113 | TAUJ1 CH3 interrupt |
| DMACTRG[9] | INTTAUJ210 | TAUJ2 CH0 interrupt |
| DMACTRG[10] | INTTAUJ211 | TAUJ2 CH1 interrupt |
| DMACTRG[11] | INTTAUJ212 | TAUJ2 CH2 interrupt |
| DMACTRG[12] | INTTAUJ213 | TAUJ2 CH3 interrupt |
| DMACTRG[13] | INTTAUD010 | TAUD0 CH0 interrupt if DMACSEL000 = 0 RX FIFO DMA request 0 if DMACSEL000 = 1 |
| DMACTRG[14] | INTTAUD011 | TAUD0 CH1 interrupt if DMACSEL001 = 0 RX FIFO DMA request 1 if DMACSEL001 = 1 |
| DMACTRG[15] | INTTAUD012 | TAUD0 CH2 interrupt if DMACSEL002 = 0 RX FIFO DMA request 2 if DMACSEL002 = 1 |
| DMACTRG[16] | INTTAUD013 | TAUD0 CH3 interrupt if DMACSEL003 = 0 RX FIFO DMA request 3 if DMACSEL003 = 1 |
| DMACTRG[17] | INTTAUD014 | TAUD0 CH4 interrupt if DMACSEL004 = 0 RX FIFO DMA request 4 if DMACSEL004 = 1 |
| DMACTRG[18] | INTTAUD015 | TAUD0 CH5 interrupt if DMACSEL005 = 0 RX FIFO DMA request 5 if DMACSEL005 = 1 |
| DMACTRG[19] | INTTAUD016 | TAUD0 CH6 interrupt if DMACSEL006 = 0 RX FIFO DMA request 6 if DMACSEL006 = 1 |
| DMACTRG[20] | INTTAUD017 | TAUD0 CH7 interrupt if DMACSEL007 = 0 RX FIFO DMA request 7 if DMACSEL007 = 1 |
| DMACTRG[21] | INTTAUD018 | TAUD0 CH8 interrupt if DMACSEL008 = 0 COM FIFO DMA request 0 if DMACSEL008 = 1 |
| DMACTRG[22] | INTTAUD019 | TAUD0 CH9 interrupt if DMACSEL009 = 0 COM FIFO DMA request 1 if DMACSEL009 = 1 |

Table 7.5 List of DMA Trigger Sources (2/4)

| Channel | Interrupt Name | DMA Trigger Source |
|-------------|----------------|--|
| DMACTRG[23] | INTTAUD0I10 | TAUD0 CH10 interrupt if DMACSEL010 = 0 COM FIFO DMA request 2 if DMACSEL010 = 1 |
| DMACTRG[24] | INTTAUD0I11 | TAUD0 CH11 interrupt |
| DMACTRG[25] | INTTAUD0I12 | TAUD0 CH12 interrupt |
| DMACTRG[26] | INTTAUD0I13 | TAUD0 CH13 interrupt |
| DMACTRG[27] | INTTAUD0I14 | TAUD0 CH14 interrupt |
| DMACTRG[28] | INTTAUD0I15 | TAUD0 CH15 interrupt |
| DMACTRG[29] | INTTAUD1I0 | TAUD1 CH0 interrupt |
| DMACTRG[30] | INTTAUD1I1 | TAUD1 CH1 interrupt |
| DMACTRG[31] | INTTAUD1I2 | TAUD1 CH2 interrupt |
| DMACTRG[32] | INTTAUD1I3 | TAUD1 CH3 interrupt |
| DMACTRG[33] | INTTAUD1I4 | TAUD1 CH4 interrupt |
| DMACTRG[34] | INTTAUD1I5 | TAUD1 CH5 interrupt if DMACSEL100 = 0 RX FIFO DMA request 0 if DMACSEL100 = 1 |
| DMACTRG[35] | INTTAUD1I6 | TAUD1 CH6 interrupt if DMACSEL101 = 0 RX FIFO DMA request 1 if DMACSEL101 = 1 |
| DMACTRG[36] | INTTAUD1I7 | TAUD1 CH7 interrupt if DMACSEL102 = 0 RX FIFO DMA request 2 if DMACSEL102 = 1 |
| DMACTRG[37] | INTTAUD1I8 | TAUD1 CH8 interrupt if DMACSEL103 = 0 RX FIFO DMA request 3 if DMACSEL103 = 1 |
| DMACTRG[38] | INTTAUD1I9 | TAUD1 CH9 interrupt if DMACSEL104 = 0 RX FIFO DMA request 4 if DMACSEL104 = 1 |
| DMACTRG[39] | INTTAUD1I10 | TAUD1 CH10 interrupt if DMACSEL105 = 0 RX FIFO DMA request 5 if DMACSEL105 = 1 |
| DMACTRG[40] | INTTAUD1I11 | TAUD1 CH11 interrupt if DMACSEL106 = 0 RX FIFO DMA request 6 if DMACSEL106 = 1 |
| DMACTRG[41] | INTTAUD1I12 | TAUD1 CH12 interrupt if DMACSEL107 = 0 RX FIFO DMA request 7 if DMACSEL107 = 1 |
| DMACTRG[42] | INTTAUD1I13 | TAUD1 CH13 interrupt if DMACSEL108 = 0 CON FIFO DMA request 0 if DMACSEL108 = 1 |
| DMACTRG[43] | INTTAUD1I14 | TAUD1 CH14 interrupt if DMACSEL109 = 0 CON FIFO DMA request 1 if DMACSEL109 = 1 |
| DMACTRG[44] | INTTAUD1I15 | TAUD1 CH15 interrupt if DMACSEL110 = 0 CON FIFO DMA request 2 if DMACSEL110 = 1 |
| DMACTRG[45] | INTTAUD2I0 | TAUD2 CH0 interrupt |
| DMACTRG[46] | INTTAUD2I1 | TAUD2 CH1 interrupt |
| DMACTRG[47] | INTTAUD2I2 | TAUD2 CH2 interrupt |
| DMACTRG[48] | INTTAUD2I3 | TAUD2 CH3 interrupt |
| DMACTRG[49] | INTTAUD2I4 | TAUD2 CH4 interrupt |
| DMACTRG[50] | INTTAUD2I5 | TAUD2 CH5 interrupt |
| DMACTRG[51] | INTTAUD2I6 | TAUD2 CH6 interrupt |
| DMACTRG[52] | INTTAUD2I7 | TAUD2 CH7 interrupt |
| DMACTRG[53] | INTTAUD2I8 | TAUD2 CH8 interrupt |
| DMACTRG[54] | INTTAUD2I9 | TAUD2 CH9 interrupt |
| DMACTRG[55] | INTADCG0I0 | ADCG0 SG0 end interrupt |
| DMACTRG[56] | INTADCG0I1 | ADCG0 SG1 end interrupt |
| DMACTRG[57] | INTADCG0I2 | ADCG0 SG2 end interrupt |
| DMACTRG[58] | INTADCG0I3 | ADCG0 SG3 end interrupt |

Table 7.5 List of DMA Trigger Sources (3/4)

| Channel | Interrupt Name | DMA Trigger Source |
|-------------|----------------|---|
| DMACTRG[59] | INTADCG0I4 | ADCG0 SG4 end interrupt |
| DMACTRG[60] | INTADCG1I0 | ADCG1 SG0 end interrupt |
| DMACTRG[61] | INTADCG1I1 | ADCG1 SG1 end interrupt |
| DMACTRG[62] | INTADCG1I2 | ADCG1 SG2 end interrupt |
| DMACTRG[63] | INTADCG1I3 | ADCG1 SG3 end interrupt |
| DMACTRG[64] | INTADCG1I4 | ADCG1 SG4 end interrupt |
| DMACTRG[65] | INTCSIH0IR0S | CSIH0 receive status/ CS0 receive status interrupt |
| DMACTRG[66] | INTCSIH0IC0S | CSIH0 communication status/ CS0 communication status interrupt |
| DMACTRG[67] | INTCSIH0IR1 | CSIH0 CS1 receive status interrupt |
| DMACTRG[68] | INTCSIH0IC1 | CSIH0 CS1 communication status interrupt |
| DMACTRG[69] | INTCSIH0IR2 | CSIH0 CS2 receive status interrupt |
| DMACTRG[70] | INTCSIH0IC2 | CSIH0 CS2 communication status interrupt |
| DMACTRG[71] | INTCSIH0JC | CSIH0 JOB completion interrupt |
| DMACTRG[72] | INTCSIH1IR0S | CSIH1 receive status/ CS0 receive status interrupt |
| DMACTRG[73] | INTCSIH1IC0S | CSIH1 communication status/ CS0 communication status interrupt |
| DMACTRG[74] | INTCSIH1IR1 | CSIH1 CS1 receive status interrupt |
| DMACTRG[75] | INTCSIH1IC1 | CSIH1 CS1 communication status interrupt |
| DMACTRG[76] | INTCSIH1IR2 | CSIH1 CS2 receive status interrupt |
| DMACTRG[77] | INTCSIH1IC2 | CSIH1 CS2 communication status interrupt |
| DMACTRG[78] | INTCSIH1JC | CSIH1 JOB completion interrupt |
| DMACTRG[79] | INTCSIH2IR | CSIH2 receive status |
| DMACTRG[80] | INTCSIH2IC | CSIH2 communication status |
| DMACTRG[81] | INTCSIH2JC | CSIH2 JOB completion interrupt |
| DMACTRG[82] | INTCSIH3IR | CSIH3 receive status |
| DMACTRG[83] | INTCSIH3IC | CSIH3 communication status |
| DMACTRG[84] | INTCSIH3JC | CSIH3 JOB completion interrupt |
| DMACTRG[85] | INTCSIG0IR | CSIG0 receive status interrupt |
| DMACTRG[86] | INTCSIG0IC | CSIG0 transmit status interrupt |
| DMACTRG[87] | INTSCI30RXI | SCI30 receive data full |
| DMACTRG[88] | INTSCI30TXI | SCI30 communication data empty |
| DMACTRG[89] | INTSCI31RXI | SCI31 receive data full |
| DMACTRG[90] | INTSCI31TXI | SCI31 communication data empty |
| DMACTRG[91] | INTSCI32RXI | SCI32 receive data full |
| DMACTRG[92] | INTSCI32TXI | SCI32 communication data empty |
| DMACTRG[93] | INTADCG0MPX | ADCG0 MPX interrupt request |
| DMACTRG[94] | INTADCG1MPX | ADCG1 MPX interrupt request |
| DMACTRG[95] | INTRLIN30UR1 | RLIN30 receive completion interrupt |
| DMACTRG[96] | INTRLIN30UR0 | RLIN30 transmit interrupt |
| DMACTRG[97] | INTRLIN31UR1 | RLIN31 receive completion interrupt |
| DMACTRG[98] | INTRLIN31UR0 | RLIN31 transmit interrupt |
| DMACTRG[99] | INTTSG30I11 | TSG30 compare match interrupt 11 |

Table 7.5 List of DMA Trigger Sources (4/4)

| Channel | Interrupt Name | DMA Trigger Source |
|--------------|----------------|---|
| DMACTRG[100] | INTTSG30I12 | TSG30 compare match interrupt12 |
| DMACTRG[101] | INTTSG30IPEK | TSG30 peak interrupt |
| DMACTRG[102] | INTTSG30IVLY | TSG30 valley interrupt |
| DMACTRG[103] | INTTSG31I11 | TSG31 compare match interrupt 11 |
| DMACTRG[104] | INTTSG31I12 | TSG31 compare match interrupt 12 |
| DMACTRG[105] | INTTSG31IPEK | TSG31 peak interrupt |
| DMACTRG[106] | INTTSG31IVLY | TSG31 valley interrupt |
| DMACTRG[107] | INTENCA0I0 | ENCA0 match/capture interrupt 0 |
| DMACTRG[108] | INTENCA0I1 | ENCA0 match/capture interrupt 1 |
| DMACTRG[109] | INTENCA1I0 | ENCA1 match/capture interrupt 0 |
| DMACTRG[110] | INTENCA1I1 | ENCA1 match/capture interrupt 1 |
| DMACTRG[111] | INTTPBA0IPRD | TPBA0 Period match detection interrupt |
| DMACTRG[112] | INTTPBA0IDTY | TPBA0 Duty match detection interrupt |
| DMACTRG[113] | INTTPBA0IPAT | TPBA0 Pattern count match detection interrupt |
| DMACTRG[114] | INTTPBA1IPRD | TPBA1 Period match detection interrupt |
| DMACTRG[115] | INTTPBA1IDTY | TPBA1 Duty match detection interrupt |
| DMACTRG[116] | INTTPBA1IPAT | TPBA1 Pattern count match detection interrupt |
| DMACTRG[117] | INTPSI50RI | PSI50 receive interrupt |
| DMACTRG[118] | INTPSI51RI | PSI51 receive interrupt |
| DMACTRG[119] | INTSENT0RI | SETN0 receive interrupt |
| DMACTRG[120] | INTSENT1RI | SETN1 receive interrupt |
| DMACTRG[121] | INTSENT2RI | SETN2 receive interrupt |
| DMACTRG[122] | INTSENT3RI | SETN3 receive interrupt |
| DMACTRG[123] | INTSENT4RI | SETN4 receive interrupt |
| DMACTRG[124] | INTSENT5RI | SETN5 receive interrupt |
| DMACTRG[125] | Reserved | — |
| DMACTRG[126] | Reserved | — |
| DMACTRG[127] | INTDMAFL | Signal for DMA program command*1 |

Note 1. For details, see the *RH850/P1M-E Flash Memory User's Manual: Hardware Interface*.

7.1.5 DTS Trigger Source

The table below shows the assignment of trigger sources to DTS inputs. Some of the trigger sources are selected from two sources (primary and secondary). After reset, the primary trigger source is selected. The secondary source is software selectable by writing a configuration register. The request status of selected trigger source is maintained in DTSFSL register.

Table 7.6 List of DTS Trigger Sources (1/4)

| Channel | Interrupt Name | DTS Trigger Source |
|------------|----------------|--|
| DTSTRG[0] | INTP9 | External interrupt |
| DTSTRG[1] | INTTAUJ0I0 | TAUJ0 CH0 interrupt |
| DTSTRG[2] | INTTAUJ0I1 | TAUJ0 CH1 interrupt |
| DTSTRG[3] | INTTAUJ0I2 | TAUJ0 CH2 interrupt |
| DTSTRG[4] | INTTAUJ0I3 | TAUJ0 CH3 interrupt |
| DTSTRG[5] | INTTAUJ1I0 | TAUJ1 CH0 interrupt |
| DTSTRG[6] | INTTAUJ1I1 | TAUJ1 CH1 interrupt |
| DTSTRG[7] | INTTAUJ1I2 | TAUJ1 CH2 interrupt |
| DTSTRG[8] | INTTAUJ1I3 | TAUJ1 CH3 interrupt |
| DTSTRG[9] | INTTAUJ2I0 | TAUJ2 CH0 interrupt |
| DTSTRG[10] | INTTAUJ2I1 | TAUJ2 CH1 interrupt |
| DTSTRG[11] | INTTAUJ2I2 | TAUJ2 CH2 interrupt |
| DTSTRG[12] | INTTAUJ2I3 | TAUJ2 CH3 interrupt |
| DTSTRG[13] | INTTAUD0I0 | TAUD0 CH0 interrupt if DTSSSEL000 = 0 RX FIFO DMA request 0 if DTSSSEL000 = 1 |
| DTSTRG[14] | INTTAUD0I1 | TAUD0 CH1 interrupt if DTSSSEL001 = 0 RX FIFO DMA request 1 if DTSSSEL001 = 1 |
| DTSTRG[15] | INTTAUD0I2 | TAUD0 CH2 interrupt if DTSSSEL002 = 0 RX FIFO DMA request 2 if DTSSSEL002 = 1 |
| DTSTRG[16] | INTTAUD0I3 | TAUD0 CH3 interrupt if DTSSSEL003 = 0 RX FIFO DMA request 3 if DTSSSEL003 = 1 |
| DTSTRG[17] | INTTAUD0I4 | TAUD0 CH4 interrupt if DTSSSEL004 = 0 RX FIFO DMA request 4 if DTSSSEL004 = 1 |
| DTSTRG[18] | INTTAUD0I5 | TAUD0 CH5 interrupt if DTSSSEL005 = 0 RX FIFO DMA request 5 if DTSSSEL005 = 1 |
| DTSTRG[19] | INTTAUD0I6 | TAUD0 CH6 interrupt if DTSSSEL006 = 0 RX FIFO DMA request 6 if DTSSSEL006 = 1 |
| DTSTRG[20] | INTTAUD0I7 | TAUD0 CH7 interrupt if DTSSSEL007 = 0 RX FIFO DMA request 7 if DTSSSEL007 = 1 |
| DTSTRG[21] | INTTAUD0I8 | TAUD0 CH8 interrupt if DTSSSEL008 = 0 COM FIFO DMA request 0 if DTSSSEL008 = 1 |
| DTSTRG[22] | INTTAUD0I9 | TAUD0 CH9 interrupt if DTSSSEL009 = 0 COM FIFO DMA request 1 if DTSSSEL009 = 1 |
| DTSTRG[23] | INTTAUD0I10 | TAUD0 CH10 interrupt if DTSSSEL010 = 0 COM FIFO DMA request 2 if DTSSSEL010 = 1 |
| DTSTRG[24] | INTTAUD0I11 | TAUD0 CH11 interrupt |
| DTSTRG[25] | INTTAUD0I12 | TAUD0 CH12 interrupt |
| DTSTRG[26] | INTTAUD0I13 | TAUD0 CH13 interrupt |
| DTSTRG[27] | INTTAUD0I14 | TAUD0 CH14 interrupt |
| DTSTRG[28] | INTTAUD0I15 | TAUD0 CH15 interrupt |
| DTSTRG[29] | INTTAUD1I0 | TAUD1 CH0 interrupt |
| DTSTRG[30] | INTTAUD1I1 | TAUD1 CH1 interrupt |

Table 7.6 List of DTS Trigger Sources (2/4)

| Channel | Interrupt Name | DTS Trigger Source |
|------------|----------------|--|
| DTSTRG[31] | INTTAUD1I2 | TAUD1 CH2 interrupt |
| DTSTRG[32] | INTTAUD1I3 | TAUD1 CH3 interrupt |
| DTSTRG[33] | INTTAUD1I4 | TAUD1 CH4 interrupt |
| DTSTRG[34] | INTTAUD1I5 | TAUD1 CH5 interrupt if DTSSSEL100 = 0 RX FIFO DMA request 0 if DTSSSEL100 = 1 |
| DTSTRG[35] | INTTAUD1I6 | TAUD1 CH6 interrupt if DTSSSEL101 = 0 RX FIFO DMA request 1 if DTSSSEL101 = 1 |
| DTSTRG[36] | INTTAUD1I7 | TAUD1 CH7 interrupt if DTSSSEL102 = 0 RX FIFO DMA request 2 if DTSSSEL102 = 1 |
| DTSTRG[37] | INTTAUD1I8 | TAUD1 CH8 interrupt if DTSSSEL103 = 0 RX FIFO DMA request 3 if DTSSSEL103 = 1 |
| DTSTRG[38] | INTTAUD1I9 | TAUD1 CH9 interrupt if DTSSSEL104 = 0 RX FIFO DMA request 4 if DTSSSEL104 = 1 |
| DTSTRG[39] | INTTAUD1I10 | TAUD1 CH10 interrupt if DTSSSEL105 = 0 RX FIFO DMA request 5 if DTSSSEL105 = 1 |
| DTSTRG[40] | INTTAUD1I11 | TAUD1 CH11 interrupt if DTSSSEL106 = 0 RX FIFO DMA request 6 if DTSSSEL106 = 1 |
| DTSTRG[41] | INTTAUD1I12 | TAUD1 CH12 interrupt if DTSSSEL107 = 0 RX FIFO DMA request 7 if DTSSSEL107 = 1 |
| DTSTRG[42] | INTTAUD1I13 | TAUD1 CH13 interrupt if DTSSSEL108 = 0 CON FIFO DMA request 0 if DTSSSEL108 = 1 |
| DTSTRG[43] | INTTAUD1I14 | TAUD1 CH14 interrupt if DTSSSEL109 = 0 CON FIFO DMA request 1 if DTSSSEL109 = 1 |
| DTSTRG[44] | INTTAUD1I15 | TAUD1 CH15 interrupt if DTSSSEL110 = 0 CON FIFO DMA request 2 if DTSSSEL110 = 1 |
| DTSTRG[45] | INTTAUD2I0 | TAUD2 CH0 interrupt |
| DTSTRG[46] | INTTAUD2I1 | TAUD2 CH1 interrupt |
| DTSTRG[47] | INTTAUD2I2 | TAUD2 CH2 interrupt |
| DTSTRG[48] | INTTAUD2I3 | TAUD2 CH3 interrupt |
| DTSTRG[49] | INTTAUD2I4 | TAUD2 CH4 interrupt |
| DTSTRG[50] | INTTAUD2I5 | TAUD2 CH5 interrupt |
| DTSTRG[51] | INTTAUD2I6 | TAUD2 CH6 interrupt |
| DTSTRG[52] | INTTAUD2I7 | TAUD2 CH7 interrupt |
| DTSTRG[53] | INTTAUD2I8 | TAUD2 CH8 interrupt |
| DTSTRG[54] | INTTAUD2I9 | TAUD2 CH9 interrupt |
| DTSTRG[55] | INTADCG0I0 | ADCG0 SG0 end interrupt |
| DTSTRG[56] | INTADCG0I1 | ADCG0 SG1 end interrupt |
| DTSTRG[57] | INTADCG0I2 | ADCG0 SG2 end interrupt |
| DTSTRG[58] | INTADCG0I3 | ADCG0 SG3 end interrupt |
| DTSTRG[59] | INTADCG0I4 | ADCG0 SG4 end interrupt |
| DTSTRG[60] | INTADCG1I0 | ADCG1 SG0 end interrupt |
| DTSTRG[61] | INTADCG1I1 | ADCG1 SG1 end interrupt |
| DTSTRG[62] | INTADCG1I2 | ADCG1 SG2 end interrupt |
| DTSTRG[63] | INTADCG1I3 | ADCG1 SG3 end interrupt |
| DTSTRG[64] | INTADCG1I4 | ADCG1 SG4 end interrupt |
| DTSTRG[65] | INTCSIH0IR0S | CSIH0 receive status/ CS0 receive status interrupt |

Table 7.6 List of DTS Trigger Sources (3/4)

| Channel | Interrupt Name | DTS Trigger Source |
|-------------|----------------|---|
| DTSTRG[66] | INTCSIH0IC0S | CSIH0 communication status/ CS0 communication status interrupt |
| DTSTRG[67] | INTCSIH0IR1 | CSIH0 CS1 receive status interrupt |
| DTSTRG[68] | INTCSIH0IC1 | CSIH0 CS1 communication status interrupt |
| DTSTRG[69] | INTCSIH0IR2 | CSIH0 CS2 receive status interrupt |
| DTSTRG[70] | INTCSIH0IC2 | CSIH0 CS2 communication status interrupt |
| DTSTRG[71] | INTCSIH0JC | CSIH0 JOB completion interrupt |
| DTSTRG[72] | INTCSIH1IR0S | CSIH1 receive status/ CS0 receive status interrupt |
| DTSTRG[73] | INTCSIH1IC0S | CSIH1 communication status/ CS0 communication status interrupt |
| DTSTRG[74] | INTCSIH1IR1 | CSIH1 CS1 receive status interrupt |
| DTSTRG[75] | INTCSIH1IC1 | CSIH1 CS1 communication status interrupt |
| DTSTRG[76] | INTCSIH1IR2 | CSIH1 CS2 receive status interrupt |
| DTSTRG[77] | INTCSIH1IC2 | CSIH1 CS2 communication status interrupt |
| DTSTRG[78] | INTCSIH1JC | CSIH1 JOB completion interrupt |
| DTSTRG[79] | INTCSIH2IR | CSIH2 receive status |
| DTSTRG[80] | INTCSIH2IC | CSIH2 communication status |
| DTSTRG[81] | INTCSIH2JC | CSIH2 JOB completion interrupt |
| DTSTRG[82] | INTCSIH3IR | CSIH3 receive status |
| DTSTRG[83] | INTCSIH3IC | CSIH3 communication status |
| DTSTRG[84] | INTCSIH3JC | CSIH3 JOB completion interrupt |
| DTSTRG[85] | INTCSIG0IR | CSIG0 receive status interrupt |
| DTSTRG[86] | INTCSIG0IC | CSIG0 transmit status interrupt |
| DTSTRG[87] | INTSCI30RXI | SCI30 receive data full |
| DTSTRG[88] | INTSCI30TXI | SCI30 communication data empty |
| DTSTRG[89] | INTSCI31RXI | SCI31 receive data full |
| DTSTRG[90] | INTSCI31TXI | SCI31 communication data empty |
| DTSTRG[91] | INTSCI32RXI | SCI32 receive data full |
| DTSTRG[92] | INTSCI32TXI | SCI32 communication data empty |
| DTSTRG[93] | INTADCG0MPX | ADCG0 MPX interrupt request |
| DTSTRG[94] | INTADCG1MPX | ADCG1 MPX interrupt request |
| DTSTRG[95] | INTRLIN30UR1 | RLIN30 receive completion interrupt |
| DTSTRG[96] | INTRLIN30UR0 | RLIN30 transmit interrupt |
| DTSTRG[97] | INTRLIN31UR1 | RLIN31 receive completion interrupt |
| DTSTRG[98] | INTRLIN31UR0 | RLIN31 transmit interrupt |
| DTSTRG[99] | INTTSG30I11 | TSG30 compare match interrupt 11 |
| DTSTRG[100] | INTTSG30I12 | TSG30 compare match interrupt12 |
| DTSTRG[101] | INTTSG30IPEK | TSG30 peak interrupt |
| DTSTRG[102] | INTTSG30IVLY | TSG30 valley interrupt |
| DTSTRG[103] | INTTSG31I11 | TSG31 compare match interrupt 11 |
| DTSTRG[104] | INTTSG31I12 | TSG31 compare match interrupt 12 |
| DTSTRG[105] | INTTSG31IPEK | TSG31 peak interrupt |
| DTSTRG[106] | INTTSG31IVLY | TSG31 valley interrupt |
| DTSTRG[107] | INTENCA0I0 | ENCA0 match/capture interrupt 0 |

Table 7.6 List of DTS Trigger Sources (4/4)

| Channel | Interrupt Name | DTS Trigger Source |
|-------------|----------------|---|
| DTSTRG[108] | INTENCA0I1 | ENCA0 match/capture interrupt 1 |
| DTSTRG[109] | INTENCA1I0 | ENCA1 match/capture interrupt 0 |
| DTSTRG[110] | INTENCA1I1 | ENCA1 match/capture interrupt 1 |
| DTSTRG[111] | INTTPBA0IPRD | TPBA0 Period match detection interrupt |
| DTSTRG[112] | INTTPBA0IDTY | TPBA0 Duty match detection interrupt |
| DTSTRG[113] | INTTPBA0IPAT | TPBA0 Pattern count match detection interrupt |
| DTSTRG[114] | INTTPBA1IPRD | TPBA1 Period match detection interrupt |
| DTSTRG[115] | INTTPBA1IDTY | TPBA1 Duty match detection interrupt |
| DTSTRG[116] | INTTPBA1IPAT | TPBA1 Pattern count match detection interrupt |
| DTSTRG[117] | INTPSI50RI | PSI50 receive interrupt |
| DTSTRG[118] | INTPSI51RI | PSI51 receive interrupt |
| DTSTRG[119] | INTSENT0RI | SETN0 receive interrupt |
| DTSTRG[120] | INTSENT1RI | SETN1 receive interrupt |
| DTSTRG[121] | INTSENT2RI | SETN2 receive interrupt |
| DTSTRG[122] | INTSENT3RI | SETN3 receive interrupt |
| DTSTRG[123] | INTSENT4RI | SETN4 receive interrupt |
| DTSTRG[124] | INTSENT5RI | SETN5 receive interrupt |
| DTSTRG[125] | — | — |
| DTSTRG[126] | — | — |
| DTSTRG[127] | — | — |

7.2 Overview

7.2.1 Overview

Direct memory access (DMA) is used to access data without going through the CPU.

DMA consists of two types of DMA transfer modules: DMAC and DTS. A DMAC includes registers for storing transfer information, and a DTS stores transfer information in the dedicated RAM (DTSRAM). DMA has two 8-channel DMAC modules and one 128-channel DTS module.

In this manual, DTFR denotes the function to select among hardware DMA transfer sources for a DMAC and retain the DMA transfer request, and DTSFSL denotes the function to retain a DMA transfer request for each DTS channel. The DTFR can handle 128 types of hardware DMA transfer sources, and the DTSFSL can handle 128 types of them.

The address space that can be used for DMA transfer is a 4 GB address space represented by a 32-bit address. For information about which resource is assigned to a particular area in the 4 GB address space and which area is accessible from DMA, see **Section 4, Address Space**.

7.2.2 Term Definition

Table 7.7 shows the terms used in this section.

Table 7.7 Definition of Terms

| Term | Meaning |
|-------------------------------|--|
| DMA transfer | A general term for data transfer carried out by DMA. |
| DMA cycle | A series of actions that consist of reading an amount of data specified by the transfer size (8/16/32/64/128 bits) from the address specified by the source address and writing it to the address specified by the destination address. The first half of the DMA cycle (reading part) is called a read cycle, and the second half (writing part) is called a write cycle. |
| Hardware DMA transfer source | A trigger for a DMA transfer request given by an internal peripheral circuit. |
| Hardware DMA transfer request | A DMA transfer request generated by a hardware DMA transfer source |
| Software DMA transfer request | A DMA transfer request generated by software writing to a register. |
| DMA transfer request | A trigger to start DMA transfer with a DMAC or DTS. |
| Transfer information (TI) | The information required for DMA transfer, including the source address, destination address, transfer data size, and transfer count. The transfer information for a DTS is specifically termed as TI. |
| DTSRAM | RAM used by a DTS to store the transfer information. |
| Single transfer | DMA transfer consisting of one DMA cycle started by one DMA transfer request. |
| Block transfer 1 | DMA transfer consisting of a number of DMA cycles specified by the transfer count in the transfer information, started by one DMA transfer request. |
| Block transfer 2 | DMA transfer consisting of a number of DMA cycles specified by the address reload count in the transfer information, started by one DMA transfer request. |
| Block transfer | A general term for both block transfer 1 and block transfer 2. |
| Last transfer | The DMA cycle carried out when the transfer count in the transfer information is 1. |
| Address reload transfer | The DMA cycle carried out when the address reload count in the transfer information is 1 if the reload function 2 is used. |
| Suspension | An action of pausing DMA transfer during block transfer. You can resume DMA transfer after suspension. |
| Resume | An action of resuming suspended DMA transfer. |
| Transfer abort | An action of aborting DMA transfer in the middle. You cannot resume DMA transfer after that. |

7.3 DMA Function

7.3.1 Basic Operation of DMA Transfer

7.3.1.1 Transfer Mode

DMA has three transfer modes.

Single Transfer

One DMA cycle is executed when a DMA transfer request is accepted.

Block Transfer 1

A number of DMA cycles specified in the transfer count register are executed when a DMA transfer request is accepted.

Block Transfer 2

A number of DMA cycles specified by the address reload count are executed when a DMA transfer request is accepted. If the address reload count is larger than the value in the transfer count register, a number of DMA cycles specified in the transfer count register are executed.

7.3.1.2 Executing a DMA Cycle

DMA always executes a write cycle after a read cycle is complete. For example, if the transfer data size is 128 bits, a write cycle is executed after a read cycle for the 128-bit data is finished. A write cycle never starts in the middle of a read cycle.

7.3.1.3 Updating Transfer Information

When a DMA cycle is executed, DMA updates transfer information as follows.

Source Address and Destination Address

Transfer information will be updated as described in **Table 7.8** according to the settings for the source address and destination address and the settings in the transfer control register such as the count directions of source address and destination address and transfer data size.

Table 7.8 Updating the Source Address and the Destination Address

| Direction of Count | Transfer Data Size | Address after Update |
|--------------------|--------------------|--|
| Increment | 8 bit | (address before update) + 0000_0001 _H |
| | 16 bit | (address before update) + 0000_0002 _H |
| | 32 bit | (address before update) + 0000_0004 _H |
| | 64 bit | (address before update) + 0000_0008 _H |
| | 128 bit | (address before update) + 0000_0010 _H |
| Decrement | 8 bit | (address before update) – 0000_0001 _H |
| | 16 bit | (address before update) – 0000_0002 _H |
| | 32 bit | (address before update) – 0000_0004 _H |
| | 64 bit | (address before update) – 0000_0008 _H |
| | 128 bit | (address before update) – 0000_0010 _H |
| Fixed | — | Same as the address before update. |

When you use the reload function, a special update rule is applied other than the one described in **Table 7.8** is applied for the last transfer and the address reload transfer. For details, see **7.3.3, Reload Function**.

Transfer Count/Address Reload Count

The transfer count is decremented by one for every DMA cycle.

The address reload count is decremented by one for every DMA cycle when the reload function 2 or block transfer 2 is used. When the reload function 2 or block transfer 2 is not used, it is not updated.

When you use the reload function, a special update rule is applied for the last transfer and the address reload transfer. For details, see **7.3.3, Reload Function**.

Other Transfer Information

Other transfer information is not updated during execution of a DMA cycle.

7.3.1.4 Last Transfer and Address Reload Transfer

The last transfer means a DMA cycle executed when the value in the transfer count register, which shows the remaining number of transfers, is one. The last transfer differs in operation compared to other DMA cycles as follows.

- The transfer completion flag (DCSTn.TC) is set when the last transfer is complete. (Only applicable for a DMAC)
- The channel operation enable (DCENn.DTE) bit is cleared when the last transfer is complete. (Only applicable for a DMAC. When the continuous transfer is disabled.)
- When the transfer completion interrupt output enable is set, a transfer completion interrupt is output when the last transfer is complete.
- When the reload function 1 is enabled, the reload function 1 is executed at the timing of the last transfer. For details, see **7.3.3, Reload Function**.

The address reload transfer means a DMA cycle executed when the reload function 2 is enabled and the address reload count is one. The reload function 2 is executed during the address reload transfer. For details, see **7.3.3, Reload Function**.

7.3.1.5 Transfer Completion Interrupt and Transfer Count Match Interrupt Outputs

DMA can output a transfer completion interrupt and a transfer count match interrupt.

Transfer Completion Interrupt Output

When the transfer completion interrupt output enable (DTCTn.TCE) is set in the transfer control register, a DMAC requests a DMAC transfer completion interrupt when the last transfer is complete.

When the transfer completion interrupt output enable (DTTCTm.TCE) is set in the transfer control register, a DTS requests a DTS transfer completion interrupt when the last transfer is complete.

Transfer Count Match Interrupt Output

When the transfer count match interrupt enable (DTCTn.CCE) is set in the transfer control register, a DMAC requests a DMAC transfer count match interrupt when the DMA cycle in which the transfer count compare register and the transfer count have the same value is complete.

When the transfer count match interrupt enable (DTTCTm.CCE) is set in the transfer control register, a DTS requests a DTS transfer count match interrupt at the completion of the DMA cycle in which the transfer count compare register and the transfer count have the same value.

Figure 7.1 shows the operation of the transfer completion interrupt and the transfer count match interrupt.

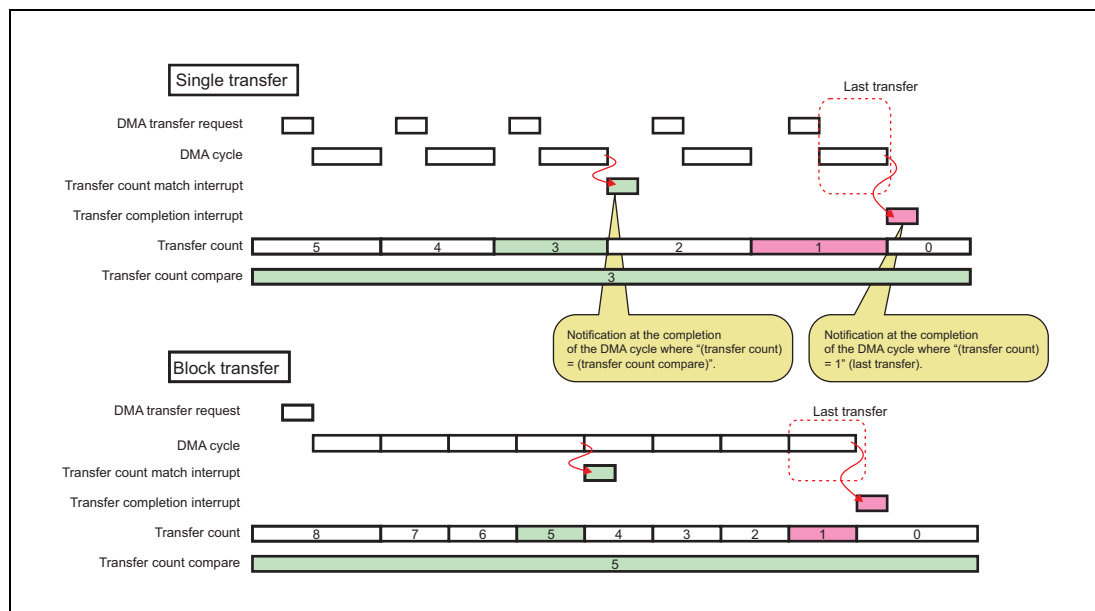


Figure 7.1 Transfer Completion Interrupt and Transfer Count Match Interrupt

7.3.1.6 Continuous Transfer

If the continuous transfer is not used, a DMAC sets the transfer completion flag (DCSTn.TC) and clears the channel operation enable (DCENn.DTE) bit when the last transfer is complete. In this case, a DMA transfer request is not accepted when the request is generated after the completion of the last transfer.

If the continuous transfer is used, the channel operation enable (DCENn.DTE) bit is not cleared when the last transfer is complete, and a DMA transfer request can be accepted even when the transfer completion flag is set. If DMA is used for a case where a specified number of DMA transfers are executed repetitively, software overhead associated with clearing the transfer completion flag and setting the channel operation enable bit after the completion of the last transfer can be reduced by using the continuous transfer.

The continuous transfer is enabled by setting the continuous transfer enable (DTCTn.MLE) in the DMAC transfer control register.

The continuous transfer is designed to work with the reload function 1. The continuous transfer function itself does not update the source address register, destination address register, and transfer count register. If, after the last transfer is complete, you want to restore the source address register, destination address register, and transfer count register to the state before the DMA transfer starts, set the values of source address register, destination address register, and transfer count register before the DMA transfer starts to the reload source address register, reload destination address register, and reload transfer count register respectively, and use reload function 1.

A DTS does not have a setting corresponding to the continuous transfer enable (DTCTn.MLE) for a DMAC. This is because a DTS does not have bits like the transfer completion flag (DCSTn.TC) and the channel operation enable (DCENn.DTE) a DMAC has.

A DTS does not start DMA transfer when a DMA transfer request is generated while the transfer count is 0. (This corresponds to the case for a DMAC where the continuous transfer is not used.)

If the reload function 1 is used for a DTS and the value other than 0 is reloaded to the transfer count when the last transfer is complete, DMA transfer can start when the next DMA transfer request is accepted. (This corresponds to the case for a DMAC where the continuous transfer is used.)

Figure 7.2 shows an operation of continuous transfer by a DMAC.

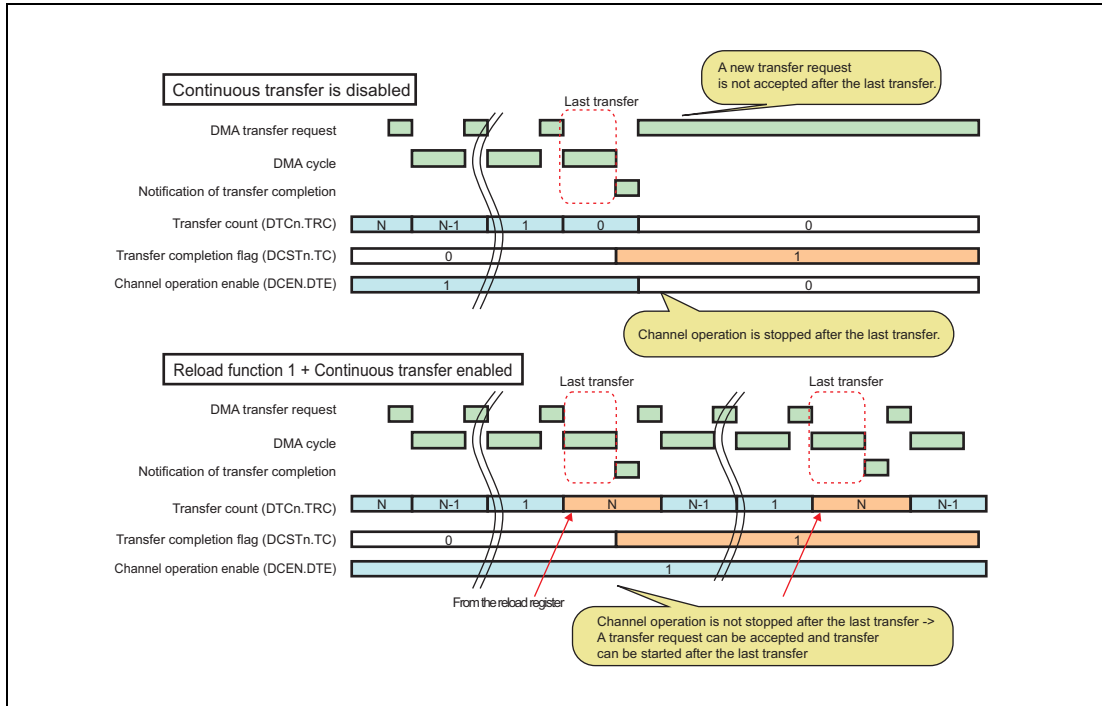


Figure 7.2 Operation of Continuous Transfer by a DMAC

7.3.2 Channel Priority Order

This subsection explains arbitration between multiple DMA channels.

7.3.2.1 DMAC Channel Arbitration

A DMAC select one channel out of eight channels with arbitration.

Arbitration is done according to the fixed priority order. The priority order is “channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7” for DMAC0, and “channel 8 > channel 9 > channel 10 > channel 11 > channel 12 > channel 13 > channel 14 > channel 15” for DMAC1.

Arbitration is done for every DMA cycle. No arbitration occurs between the read and write of a DMA cycle.

If, at the timing when one DMA cycle in the middle of a block transfer of a channel is complete, there is a DMA transfer request from a channel with a higher priority than the channel, a DMA cycle of the channel with the higher priority will be executed next as the result of arbitration.

If a DMAC executes the block transfer 1 or block transfer 2, DMAC channel arbitration is done for every DMA cycle, and possibly a DMA cycle of another DMAC channel with a higher priority can cut in.

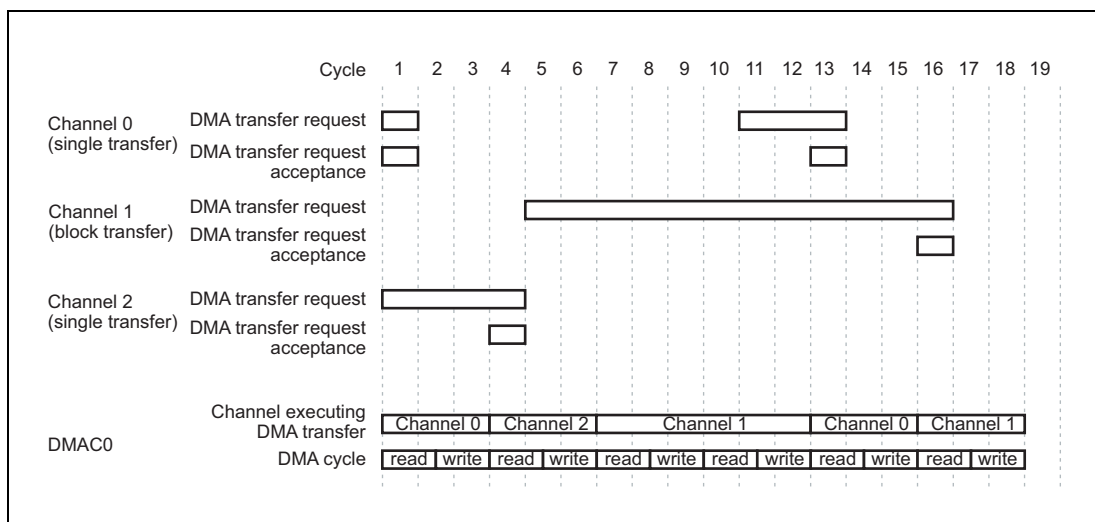


Figure 7.3 DMAC Channel Arbitration

Cycle numbers shown in **Figure 7.3** are for explanation purpose only. They do not indicate an actual number of cycles necessary for executing DMA transfer.

In **Figure 7.3**, DMA transfer requests for channels 0 and 2 are generated at Cycle 1. As a result of arbitration, a DMA cycle for channel 0 starts because its priority is higher. At Cycle 4, a DMA cycle for channel 2 starts. At Cycle 5, a DMA transfer request for channel 1 is generated. The DMA cycle for channel 2 is still ongoing and no arbitration is done at this point. At Cycle 7, a DMA cycle for channel 1 starts. Channel 1 uses block transfer. Another DMA cycle continues at Cycle 10 because there are no DMA transfer requests from other channels. At Cycle 11, a DMA transfer request for channel 0 is generated. The DMA cycle for channel 1 is still ongoing and no arbitration is done at this point. At Cycle 12, the DMA cycle for channel 1 is complete. At Cycle 13, a DMA cycle for channel 0 starts as a result of arbitration between DMA channels 0 and 1.

It should be noted that, even though a block transfer of channel 1 has been already started, a DMA cycle of not channel 1 but channel 0 is executed at Cycle 13 because the priority of the latter is higher. At Cycle 15, the DMA cycle for channel 0 is complete. At Cycle 16, a DMA cycle for channel 1 starts again. At Cycle 18, the last DMA cycle of the block transfer of channel 1 is complete.

7.3.2.2 DTS Channel Arbitration

If there are DMA transfer requests from multiple DTS channels, the DTSFSL arbitrates those DTS channels. For each DTS channel, a priority can be selected from four levels using DTS channel priority setting registers.

If there are DMA transfer request from multiple DTS channels, the arbitration is done as follows.

1. A channel with a higher priority level in the setting of DTS channel priority setting registers has a priority.
2. If two channels have the same priority level in the setting of DTS channel priority setting registers, a channel with a lower channel number has a priority.

The DTSFSL sends the DTS a DMA transfer request for the channel selected by arbitration. The DTS executes DMA transfer when it accepts the DMA transfer request.

Unlike DMA transfer with a DMAC, DMA transfer with a DTS does not allow arbitration between DTS channels in the middle of a block transfer. That means, even if a DMA transfer request with a higher priority comes during a block transfer for a channel with a lower priority, the DMA transfer with a higher priority does not start until the current block transfer for the channel with a lower priority is complete*1.

Note 1. Block transfer is complete when the last transfer for the block transfer 1 or the last transfer or address reload transfer for the block transfer 2 occurs.

When a DTS executes the block transfer 1 or block transfer 2, a DMA cycle of a DTS channel with a higher priority does not take over the ongoing block transfer until the last transfer.

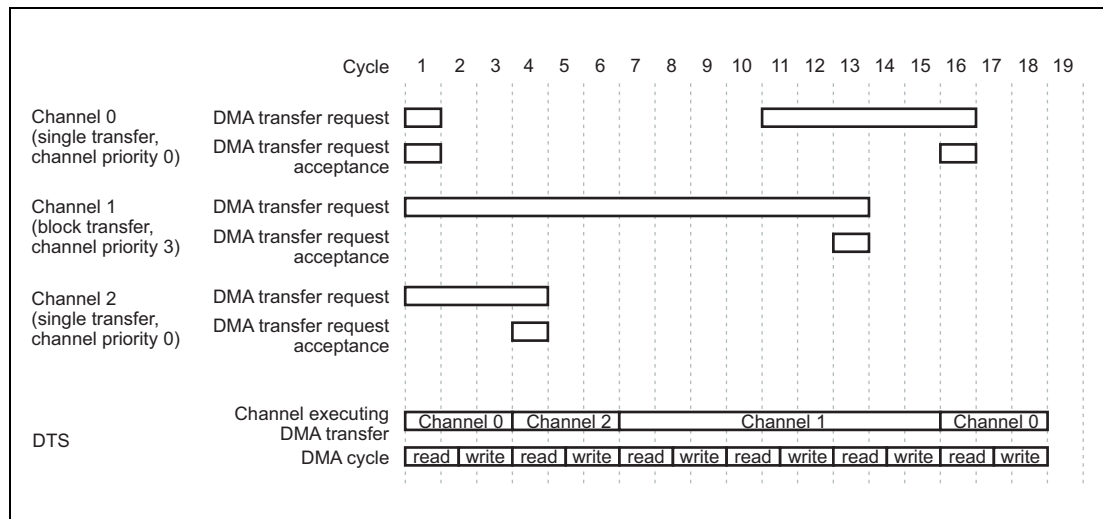


Figure 7.4 DTS Channel Arbitration

Cycle numbers shown in **Figure 7.4** are for explanation purpose only. They do not indicate an actual number of cycles necessary for executing DMA transfer.

In **Figure 7.4**, DMA transfer requests for channels 0, 1 and 2 are generated at Cycle 1. The channel priority for channels 0 and 2 is 0 and is higher than the channel priority for channel 1, which is 3. In addition, if two channels have the same priority, the channel with the smaller channel number has a higher priority. Consequently, the priority order for arbitration is “channel 0 > channel 2 > channel 1”, and a DMA cycle for channel 0 starts because its priority is the highest. At Cycle 4, as a result of arbitration between channels 1 and 2, a DMA cycle for channel 2 starts. At Cycle 7, a DMA cycle for channel 1 starts. Channel 1 uses block transfer. Another DMA cycle continues at Cycle 10 because there are no DMA transfer requests from other channels. At Cycle 11, a DMA transfer request for channel 0 is generated. The DMA cycle for channel 1 is still ongoing and no arbitration is done until the block transfer of channel 1 is complete.

At Cycle 15, the block transfer of channel 1 is complete. At Cycle 16, a DMA cycle for channel 0 starts.

7.3.2.3 Interface Arbitration

The interface between the DMA and the System Interconnect is shown in **Figure 7.5**. DMAC0, DMAC1 and DTS operate independently and they can request bus transfers via the interface arbiter to the system interconnect. As the system interconnect performs non-blocking data transfers, multiple DMA transfers can be active coincidentally.

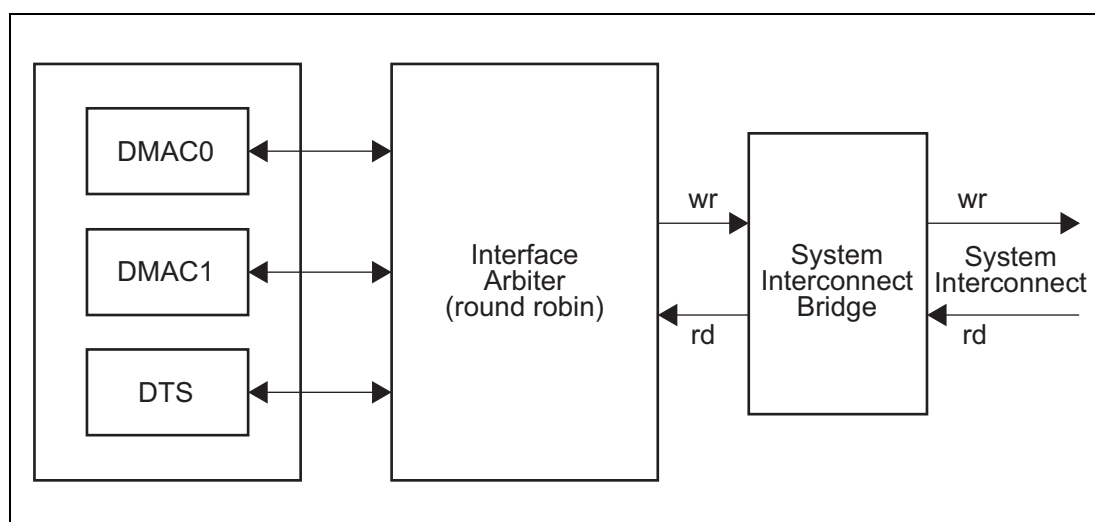


Figure 7.5 Interface Arbitration

Read and write busses are separated and they operate independently. In case of conflicts, the arbitration is done in round robin. Issuing a read or write request takes one DMA clock cycle. While the issuer waits for the response, another active unit may issue its request in the subsequent clock cycle. Note that one DMA clock cycle may take two CPU cycles.

7.3.3 Reload Function

7.3.3.1 Overview of the Reload Function

The reload function updates a portion of transfer information, more specifically, the source address, destination address, transfer count, and address reload count, to the predefined values during DMA transfer.

The reload function has two types of functions: reload function 1 and reload function 2.

7.3.3.2 Operation of Reload Function 1

When the reload function 1 is enabled, actions described in **Table 7.9** are executed at the timing of the last transfer according to the reload function 1 setting.

Table 7.9 Operation of Reload Function 1

| Reload Function 1 Setting (DTCTn.RLD1M[1:0]) | Register | Action at the Last Transfer |
|--|----------------------|---|
| 00 (Reload function 1 disabled.) | Source address | Not reloaded. |
| | Destination address | Not reloaded. |
| | Transfer count | Not reloaded. |
| | Address reload count | Not reloaded. |
| 01 (Reload function 1 enabled. Reloading source address and transfer count.) | Source address | The reload source address is copied to this. |
| | Destination address | Not reloaded. |
| | Transfer count | The reload transfer count is copied to this. |
| | Address reload count | <ul style="list-style-type: none"> If the reload function 2 is disable: Not reloaded. If the reload function 2 is enabled: The reload address reload count is copied to this. |
| 10 (Reload function 1 enabled. Reloading destination address and transfer count.) | Source address | Not reloaded. |
| | Destination address | The reload destination address is copied to this. |
| | Transfer count | The reload transfer count is copied to this. |
| | Address reload count | <ul style="list-style-type: none"> If the reload function 2 is disable: Not reloaded. If the reload function 2 is enabled: The reload address reload count is copied to this. |
| 11 (Reload function 1 enabled. Reloading source address, destination address, and transfer count.) | Source address | The reload source address is copied to this. |
| | Destination address | The reload destination address is copied to this. |
| | Transfer count | The reload transfer count is copied to this. |
| | Address reload count | <ul style="list-style-type: none"> If the reload function 2 is disable: Not reloaded If the reload function 2 is enabled: The reload address reload count is copied to this. |

Figure 7.6 shows an operation of the reload function 1.

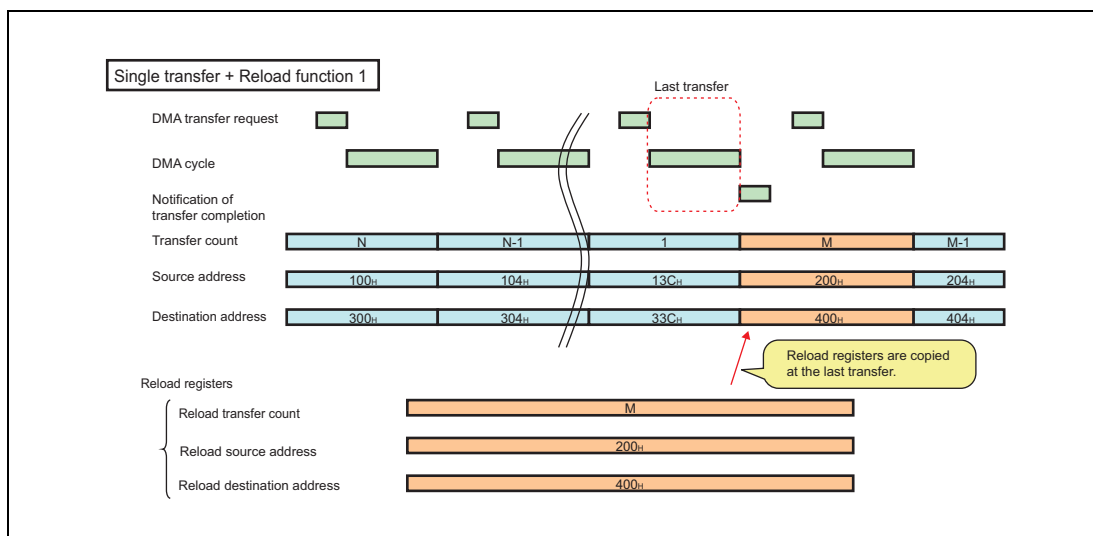


Figure 7.6 Operation of Reload Function 1

7.3.3.3 Reload Function 2

When the reload function 2 is enabled, actions described in Table 7.10 are executed at the timing of the address reload transfer according to the reload function 2 setting.

Table 7.10 Operation of Reload Function 2

| Reload Function 2 Setting (DTCTn.RLD2M[1:0]) | Register | Action at the Address Reload Transfer |
|--|----------------------|--|
| 00 (Reload function 2 disabled.) | Source address | Not reloaded. |
| | Destination address | Not reloaded. |
| | Address reload count | Not reloaded. |
| 01 (Reload function 2 enabled. Reloading source address.) | Source address | The reload source address is copied to this. |
| | Destination address | Not reloaded. |
| | Address reload count | The reload address reload count is copied to this. |
| 10 (Reload function 2 enabled. Reloading destination address.) | Source address | Not reloaded. |
| | Destination address | The reload destination address is copied to this. |
| | Address reload count | The reload address reload count is copied to this. |
| 11 (Reload function 2 enabled. Reloading source address and destination address.) | Source address | The reload source address is copied to this. |
| | Destination address | The reload destination address is copied to this. |
| | Address reload count | The reload address reload count is copied to this. |

Figure 7.7 shows an operation of the reload function 2.

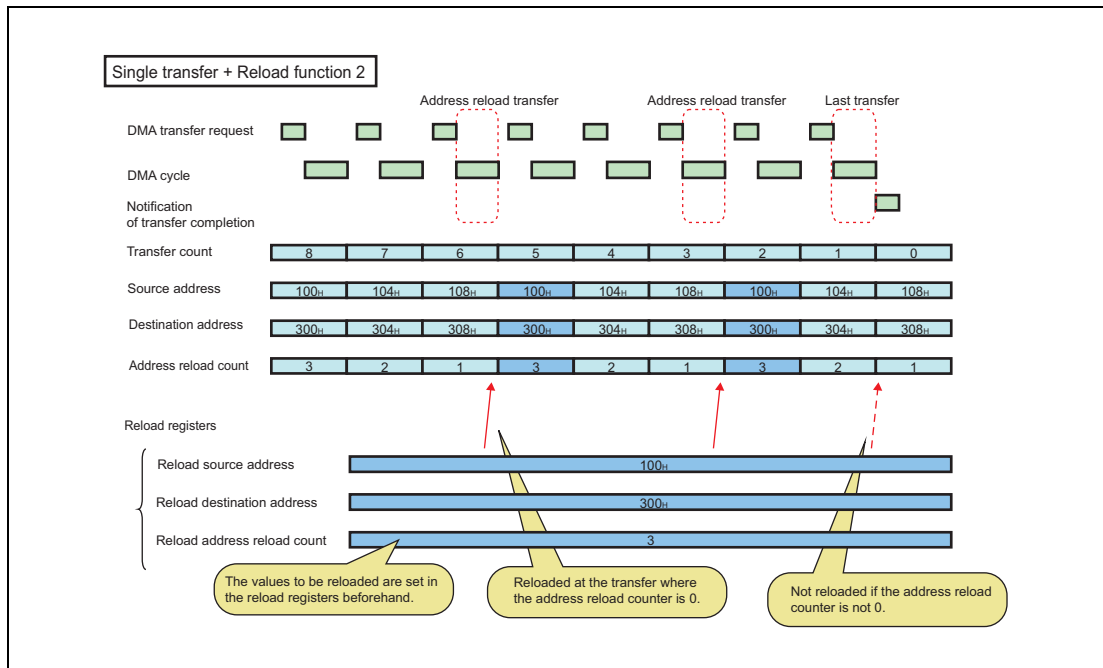


Figure 7.7 Operation of Reload Function 2

Figure 7.8 shows an operation when both the reload function 1 and the reload function 2 are used simultaneously.

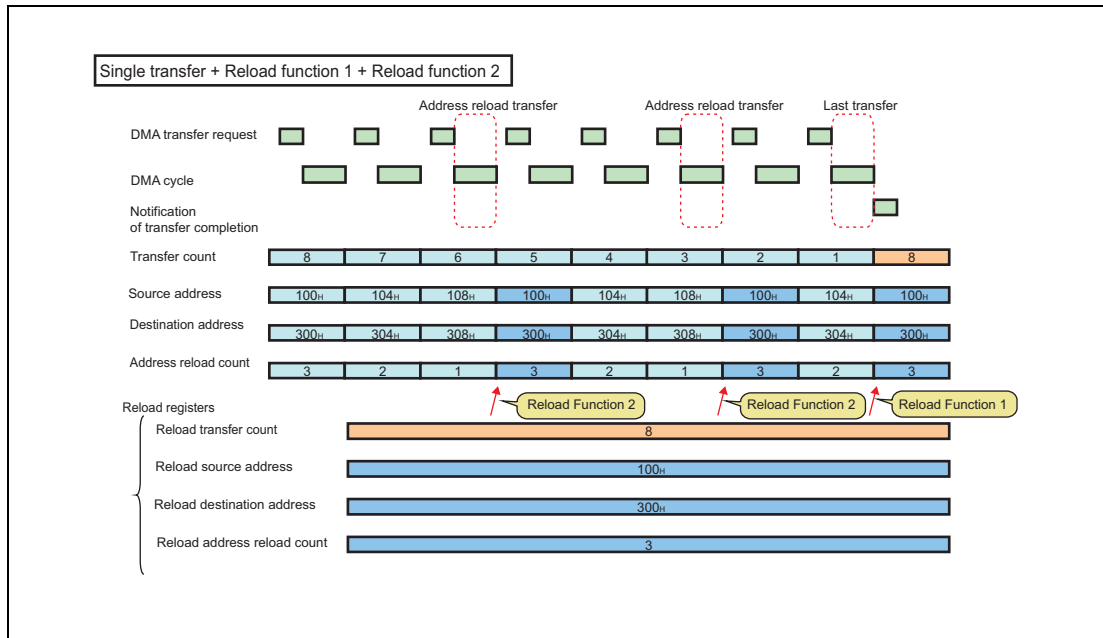


Figure 7.8 Operation when Combining the Reload Function 1 and the Reload Function 2

7.3.3.4 Timing of Setting DMAC Reload Registers

You can set up the reload source address register, reload destination address register, and reload transfer count register any time (even during DMA transfer). As an exception, if you update the reload source address register, reload destination address register, and reload transfer count register during DMA transfer, there may be a conflict between reloading at the last transfer or address reload transfer and updating the reload register. In order to avoid this conflict, setting up reload registers must be completed before the last transfer or address reload transfer starts.

If you need to update the reload source address register, reload destination address register, and reload transfer count register during DMA transfer, one way to know the right timing of update is to use a DMA transfer count match interrupt. In this case, you must set up the DMA transfer count compare register (DTCCn) so that you can have enough margin for the time necessary to update the reload registers.

7.3.3.5 Timing of Setting DTS Reload Registers

It should be noted that the right timing of setting up the reload source address information, reload destination address information, and reload transfer count information differs depending on the transfer mode.

In the single transfer mode, the TI fetched at the beginning of the last transfer or address reload transfer is used for reload at the completion of the DMA cycle. Therefore, if you use the reload function for single transfer, the reload source address information, reload destination address information, and reload transfer count information in the TI must be set up before the beginning of the last transfer or address reload transfer.

During block transfer, TI is fetched only at the beginning of DMA transfer. The TI fetched at the beginning of the DMA transfer is used for reload at the last transfer or address reload transfer. Therefore, if you use the reload function for block transfer, the reload source address information, reload destination address information, and reload transfer count information in the TI must be set up before the beginning of the DMA transfer. If you update the reload source address information, reload destination address information, and reload transfer count information in the TI in the middle of a block transfer, those new settings will not be used for reload at the completion of the block transfer.

7.3.4 Chain Function

7.3.4.1 Overview

DMA offers a function called chain function. If you use the chain function, the completion of the DMA cycle or last transfer for one channel can trigger a DMA transfer request for another channel.

A DMA transfer request for another channel initiated by the chain function is called a chain request.

You can select the condition for generating a chain request from the following two options.

- Always chain: A chain request is generated at the completion of every DMA cycle.
- Chain at the last transfer: A chain request is generated at the completion of the last transfer.

Figure 7.9 shows an operation of the case “always chain”.

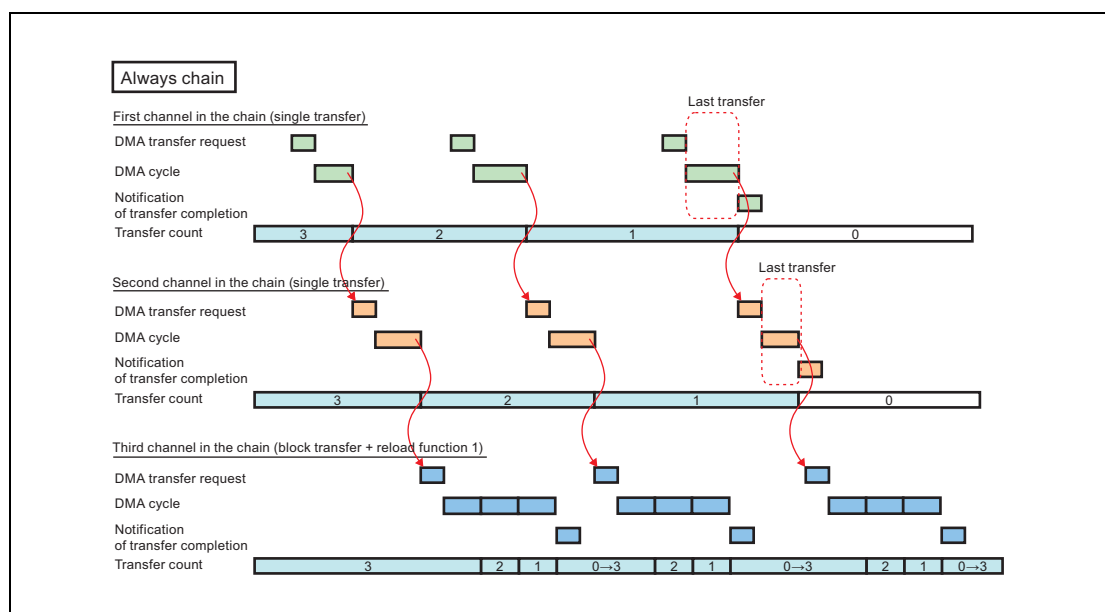


Figure 7.9 Operation of the Case “Always Chain”

Figure 7.10 shows an operation of the case “chain at the last transfer”.

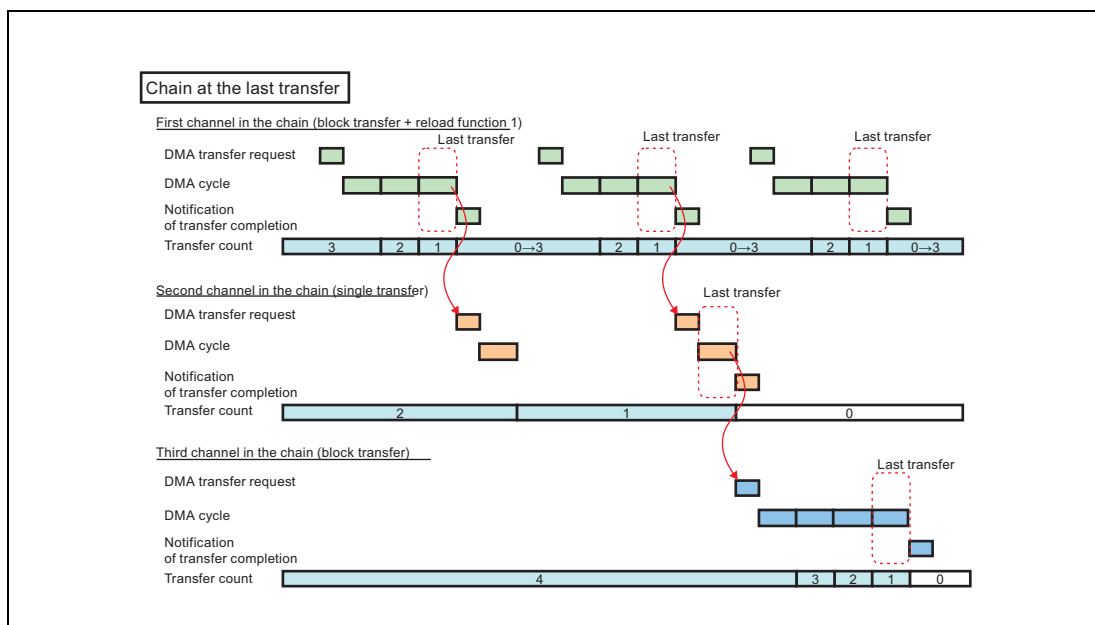


Figure 7.10 Operation of the Case “Chain at the Last Transfer”

7.3.4.2 Setting Up the Chain Function

For a DMAC, you need to write to the chain enable (DTCTn.CHNE) and the next channel in the chain selection (DTCTn.CHNSEL) in the DMAC transfer control register in order to set up the type of chain function and the next channel number in the chain.

For a DTS, you need to write to the chain enable (DTTCTm.CHNE) and the next channel in the chain selection (DTTCTm.CHNSEL) in the DTS transfer control register in order to set up the type of chain function and the next channel number in the chain.

7.3.4.3 Caution for Using the Chain Function

The chain function sets the software DMA transfer request flag of the next channel in the chain as a part of its function. Therefore, you need to set up the channel settings of the next channel in the chain in the same way as when the software DMA transfer request is used. If you specify a channel that is set up to use the hardware DMA transfer request for the next channel in the chain, the chain function does not work.

A channel and its next channel in the chain must belong to the same module (DMAC0, DMAC1, and DTS). You cannot specify a channel in another module for its next channel in the chain.

7.3.5 DMAC Operation

7.3.5.1 Types of DMA Transfer Requests and Assigning DMA Transfer Requests

A DMAC starts DMA transfer by accepting a hardware DMA transfer request or software DMA transfer request. The DMA transfer request selection assignment (DRS) bit in the DMAC transfer control register (DTCTn) determines whether a hardware DMA transfer request or a software DMA transfer request is used.

In the case of a hardware DMA transfer request for a DMAC, one out of 128 hardware DMA transfer sources is selected and assigned for each channel of the DMAC in the DTFR. This assignment is configured in the DTFR setting registers.

7.3.5.2 Generating and Accepting a Hardware DMA Transfer Request

DMAC can handle an edge-detection type of hardware DMA transfer source.

(1) Edge-detection type

In case of using edge-detection type, DTFR detects a rising edge of hardware DMA transfer source input and keeps it as a hardware DMA transfer request. DTFR also notifies DMAC of the existence of hardware DMA transfer request. When DTFR receives acknowledgement of hardware DMA transfer request from DMAC, DTFR clears hardware DMA transfer request.

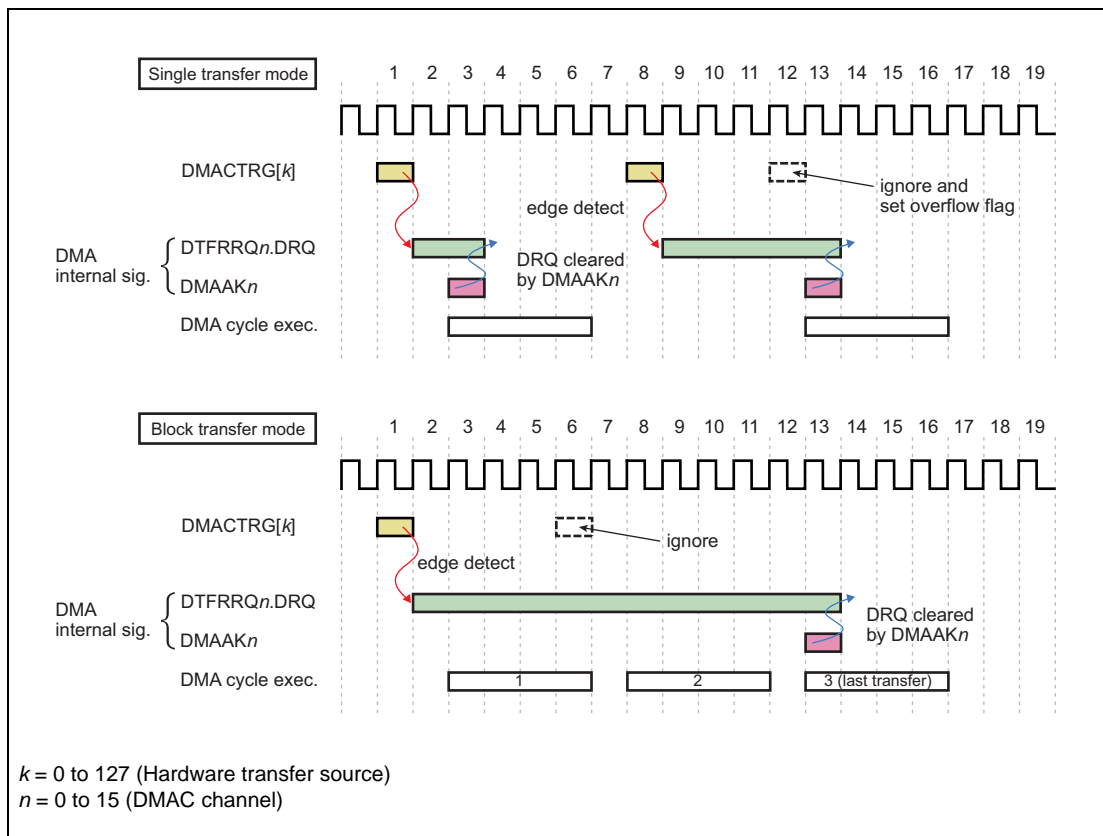


Figure 7.11 Operation of Hardware DMA Transfer Request by DMAC (Edge-detection Type)

In case of edge-detection type, DTFR can keep one hardware DMA transfer request per channel. If DTFR keeps a hardware DMA transfer request for a channel and additional rising edge of hardware

DMA transfer source is inputted for the same channel, DTFR set the hardware DMA transfer request overflow flag of corresponding channel.

(2) The Timing of Hardware DMA Transfer Acknowledgement Notification

CAUTION

Caution about the operation of DTFR hardware DMA transfer source selection enable bit during block transfer.

Be careful about the following operation when DMAC is used both with hardware DMA transfer request and with block transfer (1 or 2) mode.

If DTFR hardware DMA transfer source selection enable bit is set to disable (DTFRn.REQEN = 0) by software while DMAC is executing block transfer, the ongoing block transfer is suspended.

When DMAC is used both with hardware DMA transfer request and with block transfer (1 or 2) mode, it is recommended not to modify DTFR hardware DMA transfer source selection enable bit (DTFRn.REQEN) by software while DMAC is executing block transfer. It is also recommended to use DTCT.MLE bit to control the acknowledgement of succeeding hardware DMA transfer requests.

DTFR hardware DMA transfer request bit (DTFRRQn.DRQ) is kept set during block transfer and is cleared at the start of last transfer.(see the bottom figure of **Figure 7.12**). If DTFR hardware DMA transfer source selection enable bit is set to disable (DTFRn.REQEN = 0) by software while DMAC is executing block transfer, the hardware DMA transfer request is going to be masked and as a result DMAC cannot recognize the existence of hardware DMA transfer request.

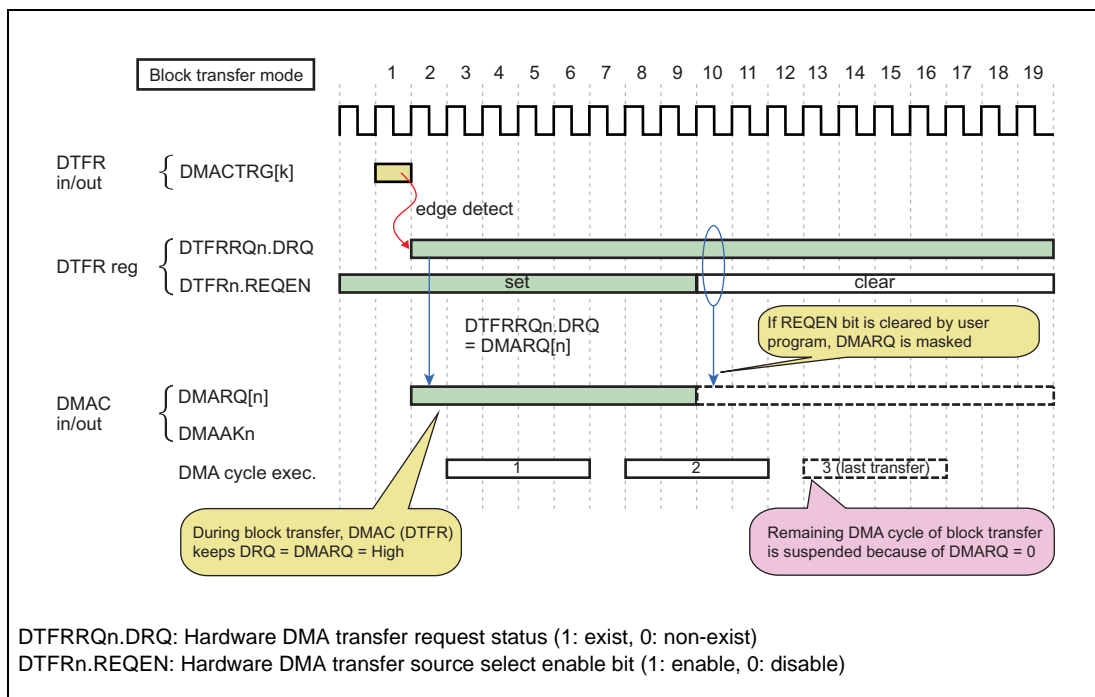


Figure 7.12 Caution about the Operation of DTFR Hardware DMA Transfer Source Selection Enable Bit during Block Transfer

This caution does not apply to the following situations.

- The case that DMAC is used both with hardware DMA transfer request and with single transfer.
- The case that DMAC is used with software DMA transfer request.
- The case that DTS is used.

7.3.5.3 Generating and Accepting a Software DMA Transfer Request

By setting the software DMA transfer request flag (SR) in the DMAC transfer status register (DCSTn) using the DMAC transfer status set register (DCSTS_n), a software DMA transfer request can be generated.

The software DMA transfer request flag is automatically cleared when the DMAC processes the DMA transfer request. The timing when the software DMA transfer request flag is automatically cleared differs depending on the transfer mode of the DMA transfer to be executed.

- In the single transfer mode, the software DMA transfer request flag is cleared whenever the software DMA transfer request is accepted.
- In the block transfer 1 mode, the software DMA transfer request flag is cleared when the last transfer starts.
- In the block transfer 2 mode, the software DMA transfer request flag is cleared when the last transfer or address reload transfer starts.

The software DMA transfer request flag can also be cleared by software using the DMAC transfer status clear register (DCSTC_n). When you abort a DMA transfer of a DMAC channel, you must clear the software DMA transfer request flag.

7.3.5.4 Execution of DMA transfer

The DMAC block executes the DMA transfer of the accepted channel. DMAC arbitration is done, when two or more channels request transfers at the same time.

Figure 7.13 shows one example of the DMAC transfer timing in single transfer mode, starting with a transfer count of 2.

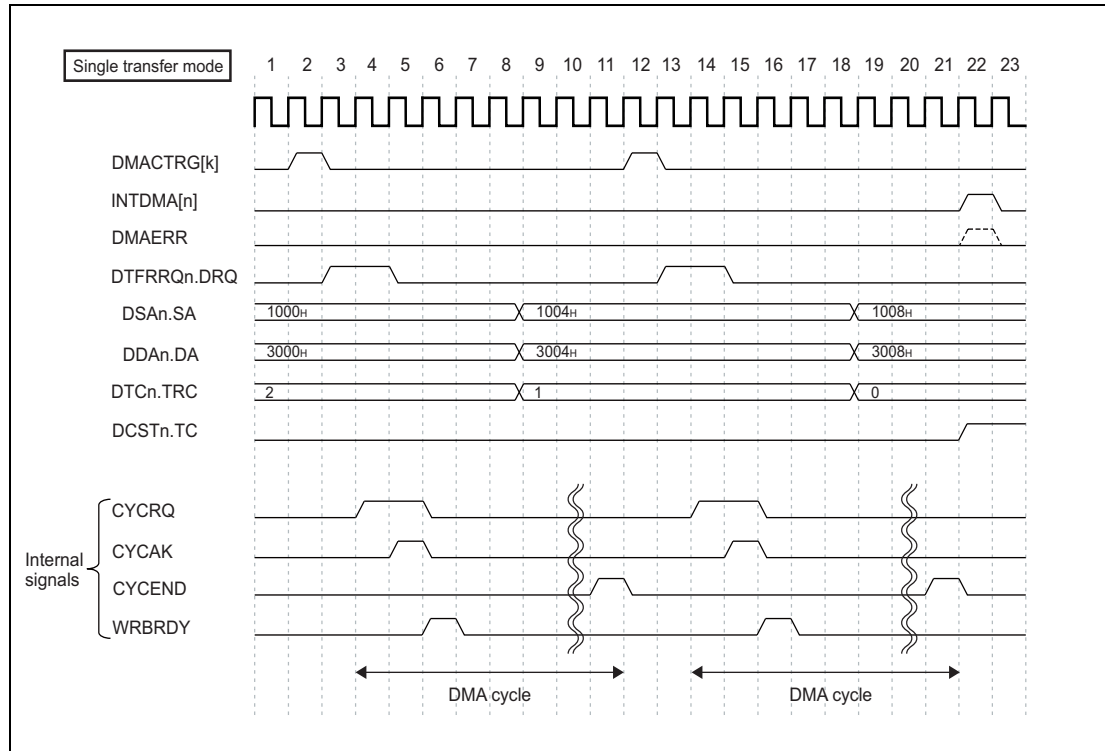


Figure 7.13 Example of DMA Transfer Timing in Single Transfer Mode

Figure 7.14 shows an example of the DMAC transfer timing in block transfer mode, starting with a transfer count of 2.

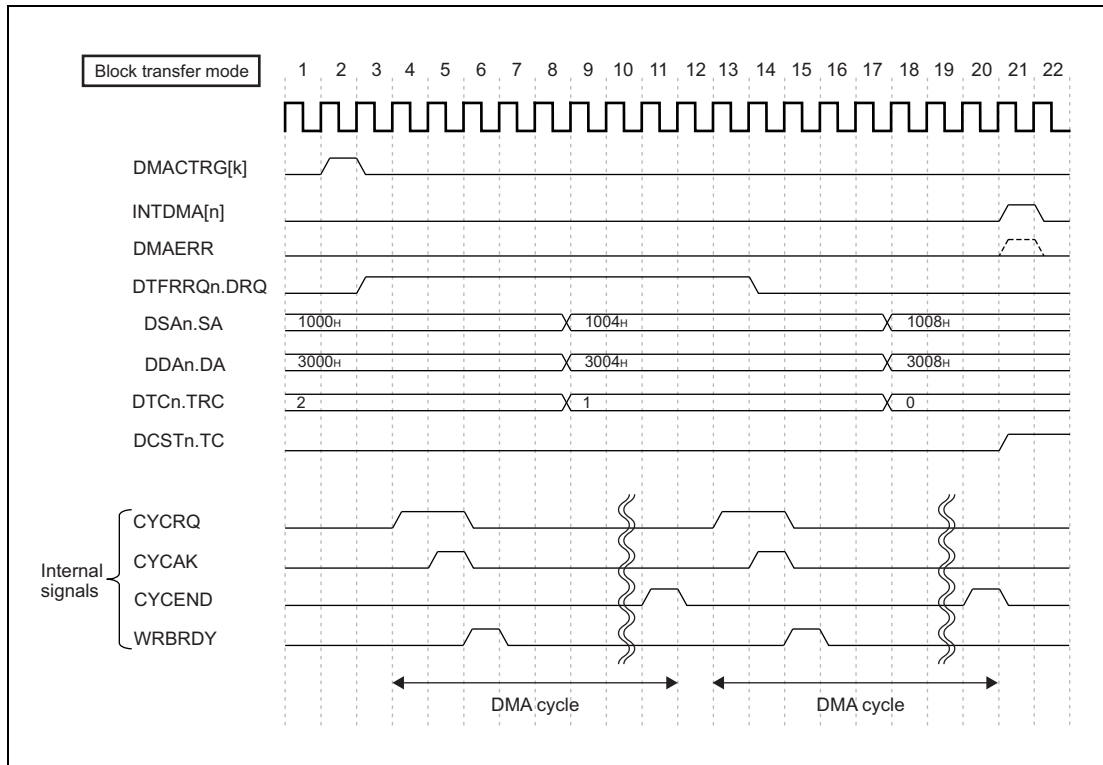


Figure 7.14 Example of DMA Transfer Timing in Block Transfer Mode

Figure 7.15 shows the timing, in which the source address register, the destination address register, the transfer count register and the transfer completion flag are updated.

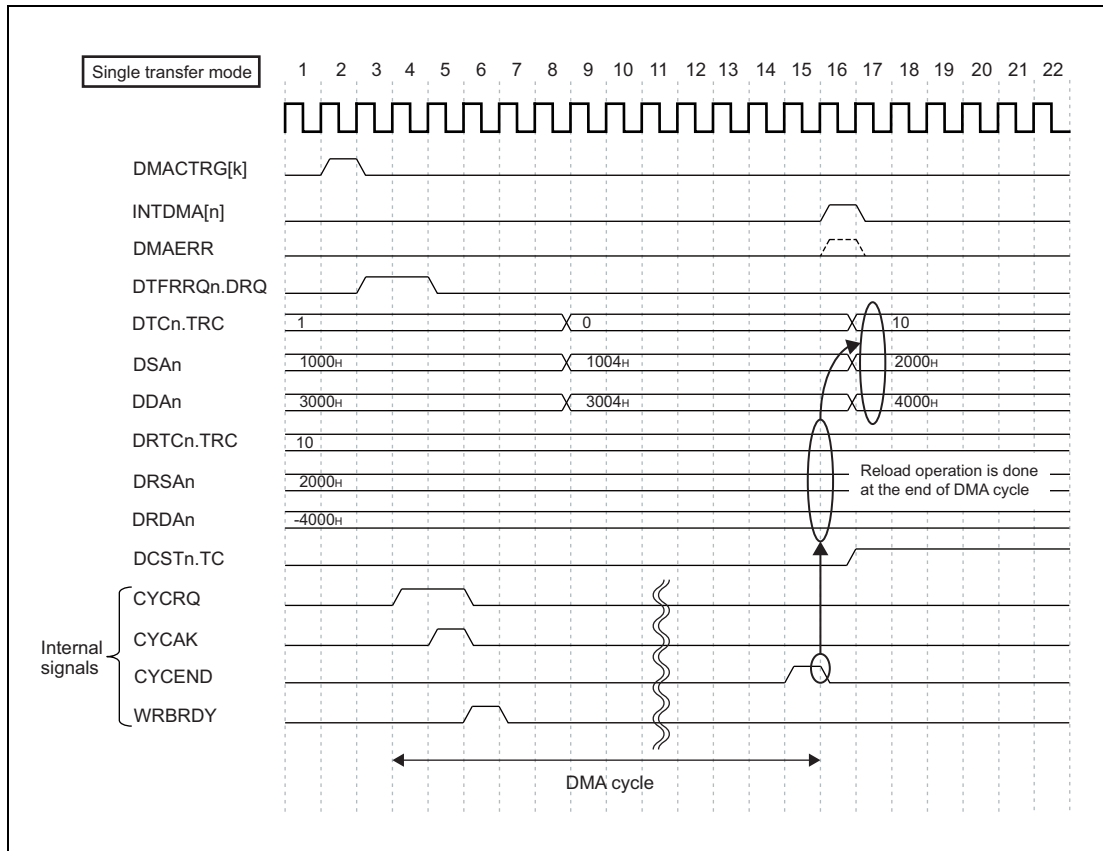


Figure 7.15 Example of DMA Transfer Timing with Reload Function 1

7.3.6 DTS Operation

7.3.6.1 Types of DMA Transfer Requests and Assigning DMA Transfer Requests

A DTS starts DMA transfer by accepting a hardware DMA transfer request or software DMA transfer request.

A transfer request for a DTS is retained in the transfer request pending bit of the DTSFSL for each channel.

As for the DTSFSL, both a hardware DMA transfer request and a software DMA transfer request are retained in the same transfer request pending bit. When executing DMA transfer, a DTS does not care whether a DMA transfer request is a hardware DMA transfer request or software DMA transfer request.

In the case of a hardware DMA transfer request for a DTS, each of 128 hardware DMA transfer sources is assigned to one of 128 channels in the DTSFSL in the fixed manner. You cannot change this assignment by, for example, register settings.

7.3.6.2 Generating and Accepting a DMA Transfer Request

When the DTSFSL detects a hardware DMA transfer source input, the DTSFSL sets the transfer request pending bit and retains the hardware DMA transfer source as a DMA transfer request. If the transfer request pending bit is set and the transfer request enable bit (DTFSLm.REQEN) in the DTSFSL operation setting register is set, the DTSFSL notifies the DTS of the DMA transfer request.

Software can also generate a DMA transfer request by setting the transfer request pending bit (DTFSTm.DRQ) of the DTSFSL transfer request status register using the DTSFSL transfer request set register (DTFSSm).

The DTSFSL can retain only one DMA transfer request per channel. If, while the transfer request pending bit for a channel is set, a new hardware DMA transfer source input for the same channel comes, DTSFSL sets request overflow flag of corresponding channel.

When the DTS accepts a DMA transfer request, it notifies of the acceptance of the DMA transfer request.

The transfer request pending bit is automatically cleared when the DTS accepts the DMA transfer request. The DTSFSL clears the transfer request pending bit automatically when the DTS accepts the DMA transfer request regardless of the type of the DMA transfer to be executed by the DTS.

The transfer request pending bit can also be cleared using the DTSFSL transfer request clear register (DTFSCm). If the transfer request pending bit of a channel is cleared before the DTS accepts the DMA transfer request, DMA transfer of the channel is not executed.

7.3.6.3 Executing DMA Transfer

When the DTS accepts a DMA transfer request for a channel, the DTS executes DMA transfer of the channel.

If there are DMA transfer requests from multiple channels, the DTSFSL arbitrates the DTS channels and picks up one channel for a DMA transfer request.

While the DTS is executing DMA transfer, the DTS transfer status (DTSSTS.DTSACT) bit in the DTS status register is set. In addition, the channel number of the currently ongoing DMA transfer is set in the DTS transfer channel (DTSSTS.DTSACH).

When the DMA transfer is complete or aborted because of DMA transfer error or writing to registers and no channel is currently executing DMA transfer, the DTS transfer status (DTSSTS.DTSACT) bit is cleared.

Figure 7.16 shows an example of a single DTS transfer. Note that the drawing shows also internal states, which are not reflected in registers. The DTS transfer is accepted in cycle 5 (ACK status), which causes fetching the transfer information (TI) in cycles 6 to 14 (TIFETCH status). The DMA cycle is executed in cycles 15 to 23 (TRANSFER status) and the TI information is written back in cycles 24 to 28 (TIWRITE status). The interval cycle is asserted in cycle 29 (FIN status). The state changes to idle (IDLE state) and the DTS waits for the next DMA request. This picture is an example and the clock cycles of each phase may differ from the depicted flow.

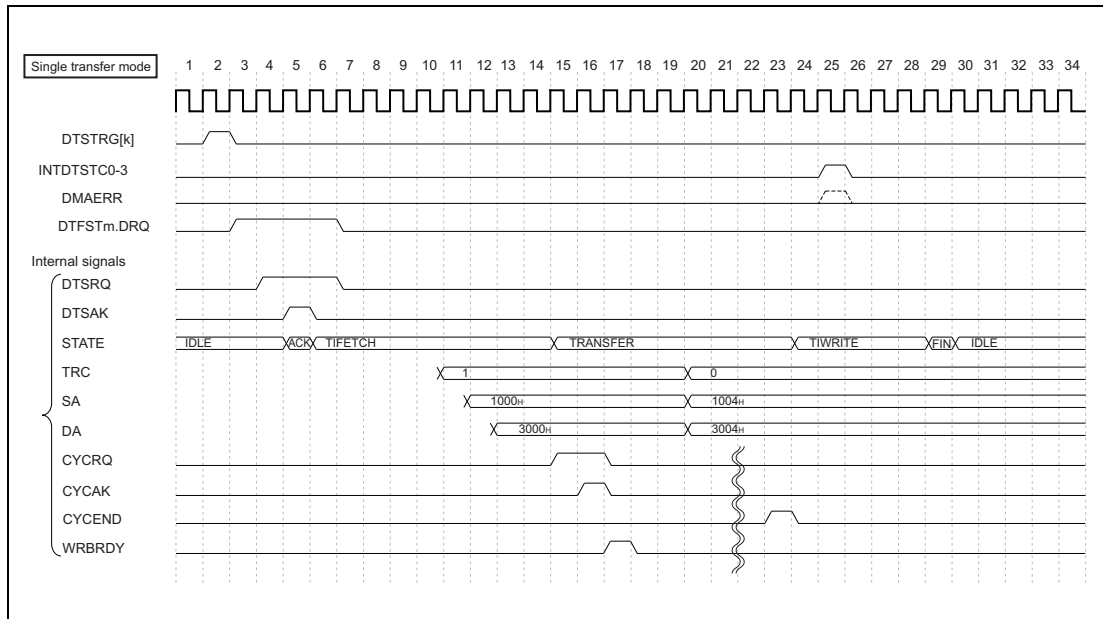


Figure 7.16 DTS Single Transfer Timing

Figure 7.17 shows an example of a DTS block transfer. The DTS transfer is accepted in cycle 5 (ACK status), which causes fetching the transfer information (TI) in cycles 6 to 10 (TIFETCH status). Three transfers are executed in cycles 11 to 25 (TRANSFER status) and the TI information is written back in cycles 26 to 30 (TIWRITE status). The interval cycle is asserted in cycle 31 (FIN status). In cycle 32, the state changes to idle (IDLE state) and the next DMA transfer can be executed.

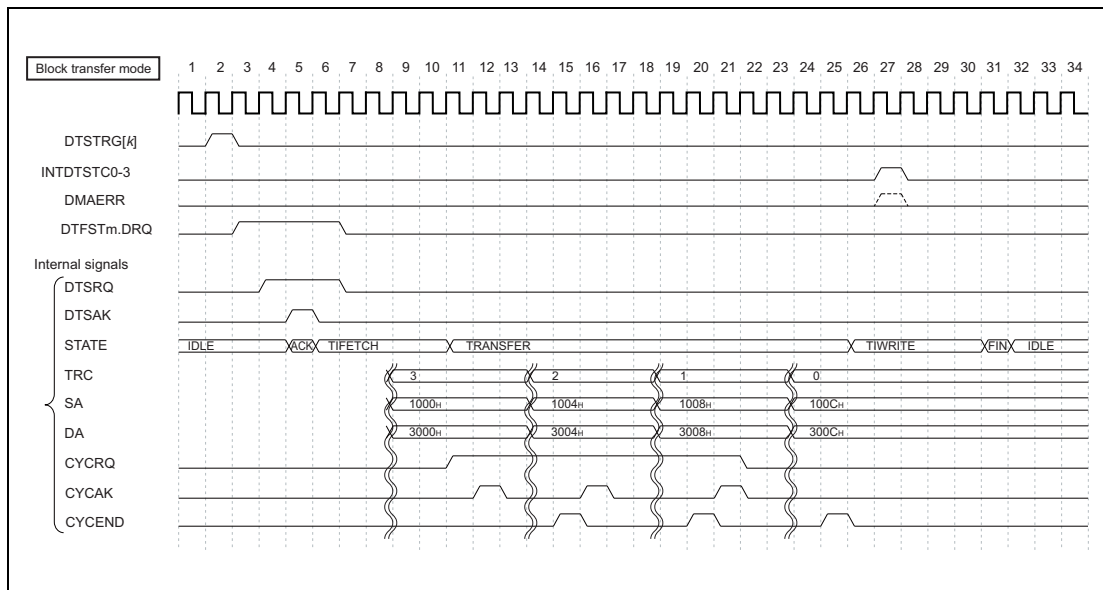


Figure 7.17 DTS Block Transfer Timing

7.3.6.4 DTSRAM Access

A DTS accesses the DTSRAM when DMA transfer starts and finishes.

A DTS's action of reading transfer information from the DTSRAM when DMA transfer starts is called TI fetch.

A DTS's action of updating the transfer information on the DTSRAM when DMA transfer finishes is called TI write back.

A single transfer performs a TI fetch at the beginning of a DMA cycle and a TI write back at the end of a DMA cycle.

A block transfer performs a TI fetch at the beginning of the first DMA cycle and a TI write back at the end of the DMA cycle that satisfies the block transfer completion condition (the last transfer or address reload transfer).

Therefore, in the case of single transfer, the transfer information on the DTSRAM is updated for each DMA cycle. In the case of block transfer, the transfer information on the DTSRAM is updated after the completion of the block transfer. If software reads the transfer information on the DTSRAM during execution of a block transfer, the transfer information at the beginning of the block transfer is read.

Table 7.11 Number of TI Fetch Cycles when not using the Chain Function

| Transfer Mode | Reload Function 1 | Reload Function 2 | Transfer Count | Address Reload Count | TI Fetch Cycles |
|---------------------|-------------------|-------------------|----------------|----------------------|-----------------|
| Single transfer | disabled | enabled | — | 1 | 12 cycles |
| | enabled | disabled | 1 | — | |
| | | enabled | >1 | 1 | — |
| | | | 1 | — | — |
| Block transfer 1 | disabled | enabled | — | Not zero | |
| | enabled | — | — | — | |
| Block transfer 2 | disabled | enabled | — | Not zero | |
| | enabled | — | — | — | |
| All other transfers | | | | | 9 cycles |

7.3.6.5 Execution Time of DTS Transfers

As shown in **Figure 7.16**, a DTS cycle is triggered by the DTSTRG signal, sampled with the rising edge of phase 3. The DTS cycle (CYCRQ) starts after TI fetch is completed. The DMAC needs two more clock cycles to issue a read and write request. The complete DTS cycle is completed after TI write back is done. TI fetch takes 9 DMA clock cycles, TI write back takes 5 DMA clock cycles.

Another two DMA clock cycles are needed to switch from write to read between the individual transfers.

In case of DTS block transfer of five words, when the DMA clock is m times slower than CPU clock, the total transfer time t_{fr_clk} for reading and writing the data by DTS is therefore:

$$t_{fr_clk} [\text{CPU clock cycle}] = (m \times 9) + (m \times 2) + 4 \times (\text{read latency} + (m \times 2) + \text{write latency} + (m \times 2)) + \text{read latency} + (m \times 2) + \text{write latency} + (m \times 1) + (m \times 5)$$

Note that the setup time for the DTS transfer (phase 3) is not considered in this example.

Note that the number of read and write latency is dependent on bus contention.

7.4 Temporarily Suspending DMA Transfers

DMA transfers can be temporarily suspended by disabling individual channels or the whole DMA controller. All DMA transfers are suspended, by setting the DMACTL.DMASPD bit. Setting the DTSTCTL1.DTSUST bit suspends DTS transfers. Each DMAC and DTS channel can hold one DMA request during executing DMA cycle, even if transfers are suspended or if the respective channel is disabled. Any additional transfer request is lost if a channel has already held a DMA request. Lost transfers due to such overflows are reported by the DTFRRQn.OVF and DTFSTm.OVF flags, again also during suspend and disable states.

7.4.1 Suspension, Restart and Abortion of a DMA Channel

DMA transfers for individual channels can be suspended by clearing the DCENn.DTE bit of that channel. A DMA cycle, which is ongoing while DTE is cleared, will be finished. DMA transfers resume, when DTE is set. As one DMA transfer request is stored during suspension, it is recommended to clear the hardware DMA transfer request in the DTFR in the case of a hardware DMA transfer request, the software DMA request flag DCSTn.SR by writing the register flag clear command into DCSTCn.SRC in the case of a software DMA transfer request, if the DMA transfer shall be entirely stopped.

Figure 7.18 shows an example of suspending, restarting and aborting DMA transfers. Both channels 0 and 1 execute block transfers. Channel 1 begins its transfer at t1 and is interrupted by the higher priority channel 0 at t2. The channel 0 transfers end at t3 and channel 1 resumes its transfer, which ends at t4. At t5, channel 1 starts the next block transfer, which is again interrupted at t6 due to the higher priority of channel 0. At t7, channel 0 is suspended and therefore channel 1 can finish its block transfer at t8. At t9, channel 0 has been restarted and it resumes its block transfer until it is again suspended at t10. At t11, the DMA channel 0 is stopped and the ongoing block transfer is aborted. Therefore, no further transfer is started at t12.

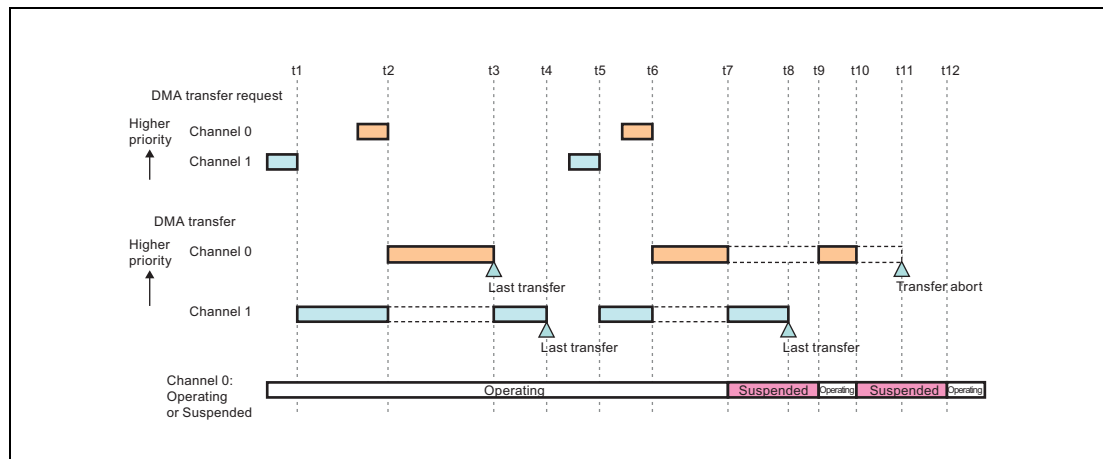


Figure 7.18 Example of Suspension, Resume, and Transfer Abort of a DMAC Channel

7.4.2 Suspension, Resume, and Transfer Abort of a DTS

You can suspend the DMA transfer executed by a DTS by setting the DTS suspend bit (DTSCCTL1.DTSUST) in the DTS control register 1. If a DMA cycle is ongoing, the DMA transfer is suspended at the timing when the DMA cycle is finished. If the ongoing DMA cycle is a single transfer or a transfer that completes a block transfer (the last transfer or address reload transfer), the DMA transfer is suspended after a TI write back after the completion of the DMA cycle. If the ongoing DMA cycle is a type other than the above, the DMA transfer is suspended after the completion of the DMA cycle without a TI write back. If you resume the DMA transfer while the DMA transfer is suspended, clear the DTS suspend bit in the DTS control register 1.

If you want to abort the currently ongoing DMA transfer executed by a DTS, suspend the DTS as described above, and then set the DTS transfer abort request bit (DTSCCTL2.DTSTIT) in the DTS control register 2 to abort the currently suspended DMA transfer. If transfer is aborted, no TI write back is executed. In addition, aborting the DMA transfer does not change the value of the DTS suspend bit (DTSCCTL1.DTSUST). If you want the DTS to accept another DMA transfer request after the abort, clear the DTS suspend bit.

Figure 7.19 shows an example of suspension, resume, and transfer abort of a DTS.

In **Figure 7.19**, channels 0, 1, and 2 are executing block transfer. At t1, a DMA transfer request for channel 1 is accepted and DMA transfer starts. At t2, DMA transfer requests for channels 0 and 2 are generated. At t3, the last transfer of channel 1 is complete, and as a result of DTS channel arbitration, a DMA transfer request for channel 0 is accepted, and DMA transfer of channel 0 starts because channel 0 has a higher priority than channel 2. At t4, the last transfer of channel 0 is complete, and DMA transfer of channel 2 starts. At t5, the DTS is put into the suspended state, and the DMA transfer of channel 2 is suspended. At t6, DMA transfer requests for channels 0 and 1 are generated. At t7, the suspended state for the DTS is cleared, and the DMA transfer of channel 2, which has been suspended in the middle of a block transfer, is resumed. If DMA transfer is suspended in the middle of a block transfer, no DTS channel arbitration is done when it is resumed. At t8, the last transfer of channel 2 is complete, and as a result of DTS channel arbitration, a DMA transfer request for channel 0 is accepted and the DMA transfer starts because channel 0 has a higher priority than channel 2. At t9, the DTS is put into the suspended state, and at t10, the suspended DMA transfer of channel 0 is aborted. When the suspended state of the DTS 0 is cleared at t11, DMA transfer of channel 1 starts because there is no currently ongoing DMA transfer and channel 1 is the only channel with a DMA transfer request.

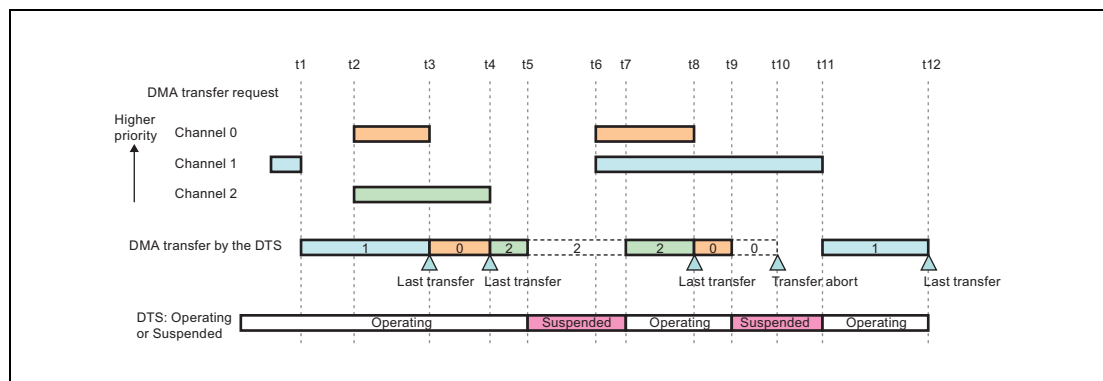


Figure 7.19 Example of Suspension, Resume, and Transfer Abort of a DTS

7.4.3 Masking and Clearing a Hardware DMA Transfer Request by the DTFR

If a DMAC uses a hardware DMA transfer request, you can temporarily disable (mask) the hardware DMA transfer request output from the DTFR to the DMAC by clearing the hardware DMA transfer source selection enable bit (DTFRn.REQEN) in the DTFR setting register.

Also, if a hardware DMA transfer source is used, you can clear a hardware DMA transfer request retained in the DTFR by using the hardware DMA transfer request clear (DTFRn.DRQC) bit in the DTFR transfer request clear register.

Even if you suspend or abort DMA transfer of a DMAC channel, the hardware DMA transfer request selection/hold circuit of the DTFR is still running, and consequently, the DTFR may retain a hardware DMA transfer request that came to the DTFR during the suspension or transfer abort period of the DMAC channel. When you resume or start DMA transfer of a DMAC channel, clear the hardware DMA transfer request retained in the DTFR as required.

Be careful that if DTFR hardware DMA transfer source selection enable bit is set to disable (DTFRn.REQEN = 0) by software while DMAC is executing block transfer, the ongoing block transfer is suspended. (refer to **7.3.5.2, Generating and Accepting a Hardware DMA Transfer Request**)

7.4.4 Masking and Clearing a Hardware DMA Transfer Request by the DTSFSL

As for a DTS, you can temporarily disable (mask) a DMA transfer request from a channel to the DTS by clearing the transfer request enable bit (DTSFSLm.REQEN) in the DTSFSL operation setting register. (The masking is actually done by excluding the channel from the candidates in DTS channel arbitration in the DTSFSL.)

Also, you can clear a DMA transfer request retained in the DTSFSL by using the transfer request clear (DTSFSLm.DRQC) bit in the DTSFSL transfer request clear register.

Regardless of the state of the DTS and the transfer request enable bit (DTSFSLm.REQEN) of the DTSFSL, the DTSFSL always monitors the hardware transfer source input from outside, and a DMA transfer request for a channel is set when a hardware transfer source for the channel is input to the DTSFSL. When you resume or start DTS transfer, clear the hardware DMA transfer request retained in the DTSFSL as required.

7.4.5 List of Suspend, Resume, and Transfer Abort Functions

Table 7.12 List of Suspend, Resume, and Transfer Abort Functions

| Function | How to execute the function | Operation | Possibility of DMA transfer abort | Master that can execute the function (See 7.6, Reliability Function.) |
|---|--|--|--|---|
| DMA suspension and resume by software control | Setting and clearing the DMACCTL.DMASPD. | All channels are in the suspended state. | Not possible* ¹ | Special master |
| Suspension and resume of a DMAC channel | Clearing and setting the DCENn.DTE in each channel register.* ² | DMA transfer of a channel is suspended. | Possible (by clearing the DMA transfer request flag during suspension) | Special master, and general master assigned to the channel. |
| Suspension and resume of a DTS | Setting and clearing the DTSCn.DTSUST. | DMA transfer of a DTS is suspended. | Possible (by setting the DTSCn.DTSTIT during suspension) | Special master |

Note 1. In order to abort DMA transfer, you need to either abort transfer for the DMAC channel or abort transfer for the DTS.

Note 2. In case that the continuous transfer enable bit (DTCTn.MLE) is set, please clear (or set) the continuous transfer enable bit (DTCTn.MLE) first.

7.5 Error Control

7.5.1 Type of Error

DMA can generate the following two types of errors.

- DMA Transfer Error

This error is generated when error is detected in the read cycle or write cycle in a DMA cycle or when error is detected by checking ECC at read cycle. This error can be generated in all DMAC and DTS channels during execution of DMA transfer.

- DTSRAM Error

This error is generated when ECC error is detected in the DTSRAM read access by a DTS. This error can be generated in the TI fetch during execution of DMA transfer for a DTS or while software is accessing the DTS channel registers.

7.5.2 DMA Transfer Error

7.5.2.1 Operation of a DMAC When DMA Transfer Error Occurs

When DMA transfer error occurs in a DMAC, the transfer error flag (DCSTn.ER) in the DMAC transfer status register of the channel with the DMA transfer error is set. The DMAC error register (DMACER) shows the transfer error flags of all 16 DMAC channels.

In a channel where the transfer error flag is set, a new DMA cycle is not executed if the transfer error case DMA transfer disable setting (DTCTn.ESE) bit is set. On the other hand, a DMA cycle is executed regardless of the value of the transfer error flag if the transfer error case DMA transfer disable setting (DTCTn.ESE) bit is cleared.

If you want to abort the DMA transfer of a channel with DMA transfer error, follow the procedure to abort DMA transfer of the DMAC channel.

If DMA transfer error occurs during the read cycle of a DMA cycle, the write cycle is not executed. If DMA transfer error occurs during the write cycle of a DMA cycle, the validity of the write is not guaranteed.

Regardless of whether DMA transfer error occurs in the read cycle or write cycle of a DMA cycle, the source address, destination address, transfer count, and address reload count registers are updated.

7.5.2.2 Operation of a DTS When DMA Transfer Error Occurs

When DMA transfer error occurs in a DTS, the DTS error flag (DTSER1.DTSER) in the DTS error register is set, and the DTS channel number with the DMA transfer error is stored in the DTS error channel (DTSER1.DTSERCH) in the same register.

If DMA transfer error occurs in a single transfer, a TI write back is executed to finish the DMA cycle.

If DMA transfer error occurs in the middle of a block transfer and the transfer error case DMA transfer abort setting (DTTCTm.ESE) is set, the remaining DMA cycles in the block transfer are not executed, but a TI write back is executed to finish the DMA cycle. At the same time, the DTS transfer status (DTSSTS.DTSACT) bit in the DTS status register is cleared. If DMA transfer error occurs in the middle of a block transfer and the transfer error case DMA transfer abort setting (DTTCTm.ESE) is cleared, the block transfer continues regardless of the DMA transfer error.

If DMA transfer error occurs during the read cycle of a DMA cycle, the write cycle is not executed. If DMA transfer error occurs during the write cycle of a DMA cycle, the validity of the write is not guaranteed.

Regardless of whether DMA transfer error occurs in the read cycle or write cycle of a DMA cycle, the source address, destination address, transfer count, and address reload count registers are updated, and the TI is updated by a TI write back.

If the DTS error flag in the DTS error register is set, a TI fetch is executed when the DTS accepts a DMA transfer request for the channel with the same channel number as the one stored in the DTS error channel. If, as a result of the TI fetch, the transfer error case DMA transfer abort setting (DTTCTm.ESE) is found to be set, a DMA cycle and a TI write back are not executed. If the transfer error case DMA transfer abort setting (DTTCTm.ESE) is cleared, DMA transfer is executed.

If the DTS error flag in the DTS error register is set, DMA transfer is executed when the DTS accepts a DMA transfer request for a channel with a channel number other than the one stored in the DTS error channel.

7.5.3 DTSRAM Error

There are two types of DTSRAM errors detected in the DTSRAM read access: ECC 1-bit error and ECC 2-bit error.

If an ECC 1-bit error is detected during a TI fetch, error corrected data is used, and DMA transfer continues. If an ECC 1-bit error is detected during DTS channel register access from software, error corrected data is returned as read data. In either case, the DTSRAM SEC error flag (DTSER2.RAMSED) in the DTS error register 2 is set, and the address of the error location in the DTSRAM is stored to the DTSRAM SEC error address (DTSER2.RAMSEAD).

In addition, the error is notified to the ECM according to DTSRAM error notification control register (DTRERINT.SEDIE).

If an ECC 2-bit error is detected during a TI fetch, handling of the DMA transfer request is terminated without executing a DMA cycle and TI write back. If an ECC 2-bit error is detected during DTS channel register access from software, peripheral bus error is notified. In either case, the DTSRAM DED error flag (DTSER2.RAMDED) in the DTS error register 2 is set, and the address of the error location in the DTSRAM is stored to the DTSRAM DED error address (DTSER2.RAMDEDAD).

In addition, the error is notified to the ECM according to DTSRAM error notification control register (DTRERINT.DEDIE).

7.5.3.1 DTSRAM ECC Test

The proper generation of the error correction code of the DTS RAM can be tested with the following algorithm:

1. Make sure that the RAM location to be tested is not modified by any ongoing DTS transfer. It is recommended to disable DTS entirely.
2. Write 0x4000_0002 to DTSRAM Test Control Register (DTRTSCTL). Enable ECC Test Mode, encode ECC from write data.
3. Write data to any DTSRAM location.
4. Read data from the same DTSRAM location.
5. Check ECC data in DTSRAM Test Reading Data Register (DTRTRDAT).
6. Restore DTS activity as required.

7.5.3.2 Stimulation of DTSRAM ECC Error

DTSRAM ECC errors can be stimulated to test the proper operation of the error control module. The following algorithm can be used to stimulate such an ECC error:

1. Make sure that the RAM location to be tested is not modified by any ongoing DTS transfer.
It is recommended to disable DTS entirely.
2. Write 0x4000_0003 to DTSRAM Test Control Register (DTRTSCTL).
Enable ECC Test Mode, write ECC from DTRTWDAT.TWDAT[6:0].
3. Write good or faulty ECC into DTRTWDAT.TWDAT[6:0].
4. Write data to any DTSRAM location.
5. Read data from the same DTSRAM location.
good ECC should not generate an ECC error.
faulty ECC should generate an ECC error.
6. Restore DTS activity as required.

7.6 Reliability Function

7.6.1 Overview

In this product, DMA is a resource used by multiple masters (PE). In order for DMA to support multi-core configuration, the following reliability functions are offered.

- Register access protection function
- Master information inherit function

7.6.2 Register Access Protection Function

This product is designed to assign each DMA channel to a CPU1 (PE).

The register access protection function allows access to the transfer information of each DMA channel from the master (PE) assigned to the channel but prohibits access from other masters.

The register access protection function prevent the channel settings from being updated by masters other than the one assigned to the channel. Please note that any master can always read all registers and that the DMAC/DTS controller cannot read or write its own registers.

7.6.2.1 Identifying the Accessing Master

DMA identifies a master based on the ID of the accessing CPU (PEID), the system protection ID configured by the accessing CPU (SPID), and the state of PSW.UM.

7.6.2.2 Special Master Access

DMA treats access from the CPU1 supervisor mode as special master access.

7.6.2.3 General Master Access

In master access, access to the following registers is allowed.

- Channel registers of the channels assigned by the channel assignment. (For details, see **7.6.2.4, Channel Assignment**.)

In general master access, write access to registers other than the above is not allowed.

7.6.2.4 Channel Assignment

To each channel, DMA can assign a master (PE) so that the master is allowed to use the channel.

Channel assignment is configured in the channel master setting register (DMxCM (x = 00 to 07, 10 to 17) in the case of a DMAC and DTSmCM in the case of a DTS) by the CPU in the supervisor mode.

In general master access, the master assigned to a channel by channel assignment is allowed to access the channel registers of the channel. If the channel registers of a channel is write accessed by a master other than the master assigned to the channel, the access is called illegal access. For information about illegal access, see **7.6.2.5, Illegal Access**.

7.6.2.5 Illegal Access

DMA handles the following access as illegal access.

- (a) General master write access to the global registers
- (b) General master write access to the channel registers of a channel by a master other than the master assigned to the channel

DMA's actions against illegal access are as follows.

For both cases (a) and (b),

- Write access is ignored.

Only for the case (b),

- The information about the illegal access is stored in a register access protection violation register.
- The DMAC0, DMAC1, and DTS have their own register access protection violation registers (DM0CMV, DM1CMV, and DTSCMV respectively).
- DMA assert illegal access notification to ECM.

Only the special master can access the register access protection violation registers. The special master can check whether illegal access has occurred by checking the register access protection violation registers periodically or checking DMA illegal access error of ECM.

In addition, it is recommended that, when a master tries to use DMA and configures transfer information in the channel registers, the master should check whether the configuration has been successfully completed without illegal access by, for example, reading back the settings.

7.6.3 Master Information Inherit Function

In this product, DMA access inherits master information that is equivalent to the master information of the master assigned to the DMA channel.

The master information that is output from DMA is as in **Table 7.13**.

Table 7.13 Master Information That Is Output from DMA

| Meaning | Value that is output from DMA |
|--------------------|--|
| UM | UM bit value in the channel master setting register |
| SPID* ¹ | Same as the SPID bit value in the channel master setting register as long as it is between 2 and 31. |
| PEID | Fixed to 4 |

Note 1. In case that register value is 0 or 1, DMA outputs 2 instead of register value.

7.6.4 Other Reliability Functions

7.6.4.1 Restriction on the Next Channel in the Chain

The reliability function limits the channels you can select as the next channel in the chain.

When you use the chain function, the channel master settings of a channel and its next channel in the chain must be the same.

The chain function is designed so that a channel and its next channel in the chain are managed by the same master.

When DMA detects that different masters are assigned to a channel and its next channel in the chain, it is deemed illegal and the chain function is suppressed. More specifically, when DMA tries to execute the chain function, DMA compares the chain master settings of the channel and its next channel in the chain, and if the settings are the same for both PEID and UM, the chain function is allowed and a chain request is sent to the next channel. If the settings are not the same for either PEID or UM, a chain request is not sent.

7.7 Setting Up DMA Transfer

7.7.1 Overview of Setting Up DMA

Table 7.14 Channel Assignment

| No. | Master that configures the setting | Description | Register | | Necessity of the setting | |
|-----|---|-------------------------------|--------------------------|---------------------------------|---|-------------|
| 1 | Special master (CPU in the supervisor mode) | Overall DMA operation setting | DTSPR0 to DTSPR7 | DTS channel priority setting | Mandatory (if a DTS is used) | |
| 2 | | | DM00CM to DM17CM | DMAC channel master setting | Mandatory (if a DMAC is used) | |
| 3 | | | DTS000CM to DTS127CM | DTS channel master setting | Mandatory (if a DTS is used) | |
| 4 | | | Status clear | DTSERC | DTS error clear register | Recommended |
| 5 | | | | CMVC | Channel protection violation clear register | Recommended |
| 6 | Master assigned to the DMAC channel | Channel setting | DSAn | DMAC source address | Mandatory | |
| 7 | | | DDAn | DMAC destination address | Mandatory | |
| 8 | | | DTCn | DMAC transfer count | Mandatory | |
| 9 | | | DTCTn | DMAC transfer control | Mandatory | |
| 10 | | | DRSAn | DMAC reload source address | Mandatory if the reload function is used | |
| 11 | | | DRDAn | DMAC reload destination address | Mandatory if the reload function is used | |
| 12 | | | DRTCn | DMAC reload transfer count | Mandatory if the reload function is used | |
| 13 | | | DTCCn | DMAC transfer count compare | Mandatory if the transfer count match interrupt is used | |
| 14 | | | | DTFRn | DTFR setting register | Mandatory |
| 15 | | | Status clear | DCSTCn | DMAC transfer status clear | Mandatory |
| 16 | | | | DTFRRQCn | DTFR transfer request clear | Recommended |
| 17 | | | Channel operation enable | DCENn | DMAC channel operation enable setting | Mandatory |
| 18 | Master assigned to the DTS channel | Channel setting | DTSAm | DTS source address | Mandatory | |
| 19 | | | DTDAm | DTS destination address | Mandatory | |
| 20 | | | DTTCm | DTS transfer count | Mandatory | |
| 21 | | | DTTCTm | DTS transfer control | Mandatory | |
| 22 | | | DTRSAm | DTS reload source address | Mandatory if the reload function is used | |
| 23 | | | DTRDAm | DTS reload destination address | Mandatory if the reload function is used | |
| 24 | | | DTRTCm | DTS reload transfer count | Mandatory if the reload function is used | |
| 25 | | | DTTCCm | DTS transfer count compare | Mandatory if the transfer count match interrupt is used | |
| 26 | | | Status clear | DTFSCm | DTSFSL transfer request clear | Recommended |
| 27 | | | Transfer request enable | DTFSLm | DTSFSL operation setting | Mandatory |

7.7.2 Setting Up the Overall DMA Operation

You need to set up the overall DMA operation before you start using DMA.

To configure the overall DMA operation, the special master (a CPU in the supervisor mode) needs to set up global registers. Global registers can be set up only by special master access. For details, see **7.6, Reliability Function**.

The following registers must be set up to configure the overall DMA operation.

- DTS channel priority setting registers (DTSPRy, y = 0 to 7)
Those registers configure the priority level of each DTS channel used for DTS channel arbitration.
- DMAC channel master setting registers (DMxCM, x = 00 to 07, 10 to 17)
- DTS channel master setting registers (DTSmCM)

Those registers configure channel assignment. (For details, see **7.6, Reliability Function**.)

If the DMAC channel master setting registers and the DTS channel master setting registers are not properly set, DMA channel setting and DMA transfer cannot be executed properly.

Also, if errors are detected in the following registers while the overall DMA operation is set up, clearing the errors is recommended.

- DTS error register 1 (DTSER1)
- DTS error register 2 (DTSER2)
- DMAC0 register access protection violation register (DM0CMV)
- DMAC1 register access protection violation register (DM1CMV)
- DTS register access protection violation register (DTSCMV)

7.7.3 Setting Up the DMA Channel Setting

The DMA channel setting defines the transfer information and transfer source for each DMAC and DTS channel.

To configure the DMA channel setting, the master assigned to each channel by the channel assignment needs to set up channel registers.

7.7.3.1 Setting Up the DMAC Channel Setting

Follow the procedure below to set up the DMAC channel setting and use the DMAC.

(1) Disabling the DMAC Channel Operation

If the channel operation enable (DTE) in the DMAC channel operation enable setting register (DCENn) is set, clear the DTE bit to disable the channel operation.

(2) Setting Up the Transfer Information

When you set up the transfer information of the DMAC, the following registers need to be set up.

- DMAC source address register (DSAn)
- DMAC destination address register (DDAn)
- DMAC transfer count register (DTCn)
- DMAC transfer control register (DTCTn)
- DMAC reload source address register (DRSAn)
- DMAC reload destination address register (DRDAn)
- DMAC reload transfer count register (DRTCn)
- DMAC transfer count compare register (DTCCn)

(3) Setting Up the DMA Transfer Request

While setting the transfer information, you need to set up the DMA transfer request selection assignment (DTCTn.DRS) bit in the DMAC transfer control register (DTCTn) to define whether the hardware or software DMA transfer request is used.

You cannot use both the hardware and software DMA transfer requests for the same channel at the same time.

If you use the hardware DMA transfer request, you need to select one source used as the hardware DMA transfer request out of 128 hardware DMA transfer sources using the hardware DMA transfer source selection (DTFRn.REQSEL) in the DTFR setting register. Also, you need to enable the hardware DMA transfer source selection (DTFRn.REQEN) in the same register.

The DTFR may retain a hardware DMA transfer request that came before the hardware DMA transfer source is selected. Clear the hardware DMA transfer request (DTFRRQn.DRQ) retained in the DTFR using the DTFR transfer request clear register (DTFRRQCn) if necessary.

If you use the software DMA transfer request, disable the hardware DMA transfer source selection (DTFRn.REQEN) in the DTFR setting register.

(4) Clearing the Transfer Status

The DMAC transfer status register (DCSTn) may retain the result of the previous DMA transfer. You need to clear the flags in the DMAC transfer status register using the DMAC transfer status clear register (DCSTCn).

(5) Enabling the DMAC Channel Operation

Set the channel operation enable (DCENn.DTE) bit in the DMAC channel operation enable setting register to enable the channel operation.

After the channel operation enable bit is set, the DMAC can accept a DMA transfer request and start DMA transfer.

7.7.3.2 Setting Up the DTS Channel Setting

Follow the procedure below to set up the DTS channel setting and use the DTS.

(1) Disabling the Transfer Request by the DTSFSL

Clear the transfer request enable (DTFSLm.REQEN) bit in the DTSFSL operation setting register of the DTS channel you want to set up the channel setting for. This procedure is not mandatory but recommended in order to prevent a DMA transfer request from being sent mistakenly to the DTS channel currently being configured.

It is also recommended to check the DTS status register (DTSSTS) and confirm that DMA transfer is not ongoing for the DTS channel currently being configured.

(2) Setting Up the Transfer Information

When you set up the transfer information of the DTS, the following registers need to be set up to configure the transfer information.

- DTS source address register (DTSAm)
- DTS destination address register (DTDAm)
- DTS transfer count register (DTTCm)
- DTS transfer control register (DTTCTm)
- DTS reload source address register (DTRSAm)
- DTS reload destination address register (DTRDAm)
- DTS reload transfer count register (DTRTCm)
- DTS transfer count compare register (DTTCCm)

(3) Setting Up the DMA Transfer Request

Unlike a DMAC, a DTS does not care whether a DMA transfer request is a hardware DMA transfer request or software DMA transfer request. A DTS has a transfer request pending bit for each channel in the DTSFSL, and both a hardware and software DMA transfer requests are retained in the same transfer request pending bit (DTFSTm.DRQ). Therefore, a DTS has no setting for selecting whether the hardware or software transfer request is used.

The DTSFSL may retain a DMA transfer request that came before the transfer information is set up. Clear the DMA transfer request (DTFSTm.DRQ) retained in the DTSFSL if necessary, using the DTSFSL transfer request clear register (DTFSCm).

(4) Enabling the Transfer Request by the DTSFSL

Set the transfer request enable (DTFSLm.REQEN) bit in the DTSFSL operation setting register to enable the DMA transfer request for the DTS channel.

After the transfer request enable bit for the DTSFSL is set, the DTS can accept a DMA transfer request and start DMA transfer.

7.8 Global Register

7.8.1 List of Global Register Address

Address = <DMAC_base> “FFFF 8000_H” + Offset address

Table 7.15 List of Global Register Address (1/2)

| Offset Address | Register Symbol | Meaning | Accessed Permission*2 | |
|-------------------|-----------------|---|-----------------------|----------------|
| | | | Special Master | General Master |
| 0000 _H | DMACTL | DMA control register | √ | × |
| 0010 _H | DTSCTL1 | DTS control register 1 | √ | × |
| 0014 _H | DTSCTL2 | DTS control register 2 | √ | × |
| 0018 _H | DTSSTS | DTS status register | √ | × |
| 0020 _H | DMACER | DMAC error register | √ | × |
| 0024 _H | DTSER1 | DTS error register 1 | √ | × |
| 0028 _H | DTSER2 | DTS error register 2 | √ | × |
| 002C _H | DTSERC | DTS error clear register | √ | × |
| 0030 _H | DM0CMV | DMAC0 register access protection violation register | √ | × |
| 0034 _H | DM1CMV | DMAC1 register access protection violation | √ | × |
| 0038 _H | DTSCMV | DTS register access protection violation register | √ | × |
| 003C _H | CMVC | Register access protection violation clear register | √ | × |
| 004C _H | TFRSTS | Transfer status register | √ | × |
| 0060 _H | DTSPR0 | DTS channel priority setting 0 | √ | × |
| 0064 _H | DTSPR1 | DTS channel priority setting 1 | √ | × |
| 0068 _H | DTSPR2 | DTS channel priority setting 2 | √ | × |
| 006C _H | DTSPR3 | DTS channel priority setting 3 | √ | × |
| 0070 _H | DTSPR4 | DTS channel priority setting 4 | √ | × |
| 0074 _H | DTSPR5 | DTS channel priority setting 5 | √ | × |
| 0078 _H | DTSPR6 | DTS channel priority setting 6 | √ | × |
| 007C _H | DTSPR7 | DTS channel priority setting 7 | √ | × |
| 0080 _H | DTRECCTL | DTSRAM ECC control register | √ | × |
| 0084 _H | DTRERINT | DTSRAM Error notification control register | √ | × |
| 0094 _H | DTRTCTL | DTSRAM test control register | √ | × |
| 0098 _H | DTRTWDAT | DTSRAM test write data register | √ | × |
| 009C _H | DTRTRDAT | DTSRAM test read data register | √ | × |
| 00A0 _H | ADECCTCL | ECConBUS address ECC test control register | √ | × |
| 00A4 _H | ADECCTDT | ECConBUS address ECC test data register | √ | × |
| 0100 _H | DM00CM | DMAC0 channel 0 channel master setting | √ | × |
| 0104 _H | DM01CM | DMAC0 channel 1 channel master setting | √ | × |
| 0108 _H | DM02CM | DMAC0 channel 2 channel master setting | √ | × |
| 010C _H | DM03CM | DMAC0 channel 3 channel master setting | √ | × |
| 0110 _H | DM04CM | DMAC0 channel 4 channel master setting | √ | × |
| 0114 _H | DM05CM | DMAC0 channel 5 channel master setting | √ | × |
| 0118 _H | DM06CM | DMAC0 channel 6 channel master setting | √ | × |
| 011C _H | DM07CM | DMAC0 channel 7 channel master setting | √ | × |
| 0120 _H | DM10CM | DMAC1 channel 0 channel master setting | √ | × |
| 0124 _H | DM11CM | DMAC1 channel 1 channel master setting | √ | × |
| 0128 _H | DM12CM | DMAC1 channel 2 channel master setting | √ | × |
| 012C _H | DM13CM | DMAC1 channel 3 channel master setting | √ | × |
| 0130 _H | DM14CM | DMAC1 channel 4 channel master setting | √ | × |

Table 7.15 List of Global Register Address (2/2)

| Offset Address | Register Symbol | Meaning | Accessed Permission*2 | |
|---|-----------------|--|-----------------------|----------------|
| | | | Special Master | General Master |
| 0134 _H | DM15CM | DMAC1 channel 5 channel master setting | √ | × |
| 0138 _H | DM16CM | DMAC1 channel 6 channel master setting | √ | × |
| 013C _H | DM17CM | DMAC1 channel 7 channel master setting | √ | × |
| 0200 _H + 4 × [DTS channel number] *1 (0200 _H - 03FC _H) | DTSmCM*1 | DTS channel m channel master setting*1 | √ | × |

Note 1. [DTS channel number] and “m” in the register symbols and meanings are numbers in the range from 000 to 127.

Note 2. √: supported x: not supported

CAUTION

- It is forbidden to write to the following registers when operation of DMA is enabled (DTE bit = 1 for each DMA channel). If you do, the correct operation is not guaranteed.
 - DMxCM
- It is forbidden to write to the following registers when operation of DTS is enabled (DTSACT bit = 1). If you do, the correct operation is not guaranteed.
 - DTSPRy
 - DTRECCTL
 - DTRERINT
 - DTRTSCTL
 - DTRTWDAT
 - DTSmCM

7.8.2 Details of Global Registers

7.8.2.1 DMACTL — DMA Control Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0000_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DMA SPD |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.16 DMACTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | DMA SPD | <p>DMA suspension</p> <p>This bit shows whether DMA transfer for all channels is suspended. If a user writes 1 to this bit, DMA transfer for all channels can be suspended. If a user writes 0 to this bit, suspension of DMA transfer for all channels can be cleared. The suspension controlled by this bit is independent from the suspension controlled by the transfer enable bit (DCENn.DTE) of each DMAC channel and the suspension setting bit (DTSUST) for a DTS. That means, if this bit is 1, all DMA transfers are suspended regardless of the values of the DTE bit of each channel and the DTSUST bit of the DTS.</p> <p>Writing to this bit does not affect the DTE bit of each channel and the DTSUST bit of the DTS.</p> <p>0: DMA suspension cleared 1: DMA suspension request/DMA suspension ongoing</p> |

7.8.2.2 DTSTCTL1 — DTS Control Register 1

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0010_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DTS UST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.17 DTSTCTL1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | DTSUST | DTS suspension This bit shows whether DMA transfer of a DTS is suspended. If a user writes 1 to this bit, DMA transfer of a DTS can be suspended. 0: DTS suspension cleared 1: DTS suspension request/DTS suspension ongoing |

7.8.2.3 DTSTCTL2 — DTS Control Register 2

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0014_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DTSTIT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.18 DTSTCTL2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | DTSTIT | DTS transfer abort request While the DTS is suspended, a user can write 1 to this bit to abort the suspended DMA transfer. When the suspended DMA transfer of a DTS is aborted, the DTSSTS.DTSACT bit is cleared to 0. The read value is always 0. |

7.8.2.4 DTSSTS — DTS Status Register

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0018_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | DTS CYC | DTSA CH6 | DTSA CH5 | DTSA CH4 | DTSA CH3 | DTSA CH2 | DTSA CH1 | DTSA CH0 | DTSA CT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.19 DTSSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 9 | Reserved | When read, the value after reset is read. |
| 8 | DTSCYC | DMA cycle execution state This bit shows whether a DMA cycle is ongoing in the DTS. 0: DMA cycle is not ongoing. 1: DMA cycle is ongoing. After suspending DTS transfers, DTSCYC may be polled to assure that any possibly ongoing DTS cycle has finished. |
| 7 to 1 | DTSACH[6:0] | DTS transfer channel If there is a channel in the DTS executing DMA transfer, the channel number is shown. If there is no channel in the DTS executing DMA transfer, the channel number of the last DMA transfer is shown. |
| 0 | DTSACT | DTS transfer status This bit shows whether there is a channel in the DTS executing DMA transfer. 0: There is a channel in the DTS executing DMA transfer. 1: There is no channel in the DTS executing DMA transfer. DTSACT is asserted during TI fetch, all DMA cycles and TI write-back. It is cleared after the last cycle of the TI write back is completed. If the DTS is put into the suspended state while there is a channel executing DMA transfer, this bit remains 1. If a DTS transfer abort request is made using the DTSTL2.DTSTIT bit, the suspended DTS transfer is aborted, and this bit is cleared to 0. When DMA transfer error occurs and the DMA transfer is aborted, this bit is cleared. |

7.8.2.5 DMACER — DMAC Error Register

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0020_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DM1 ER7 | DM1 ER6 | DM1 ER5 | DM1 ER4 | DM1 ER3 | DM1 ER2 | DM1 ER1 | DM1 ER0 | DM0 ER7 | DM0 ER6 | DM0 ER5 | DM0 ER4 | DM0 ER3 | DM0 ER2 | DM0 ER1 | DM0 ER0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.20 DMACER Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. |
| 15 to 8 | DM1ER[7:0] | DMAC1 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC1. Each bit is mapped from the DCSTn.ER bit of each channel of the DMAC1 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated |
| 7 to 0 | DM0ER[7:0] | DMAC0 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC0. Each bit is mapped from the DCSTn.ER bit of each channel of the DMAC0 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated |

7.8.2.6 DTSER1 — DTS Error Register 1

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0024_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----|----|----|----|----|----|-------------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | DTSER CH6 | DTSER CH5 | DTSER CH4 | DTSER CH3 | DTSER CH2 | DTSER CH1 | DTSER CH0 | — | — | — | — | — | — | DTSER WR | DTSER |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.21 DTSER1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 15 | Reserved | When read, the value after reset is read. |
| 14 to 8 | DTSERCH[6:0] | DTS error channel These bits show the DTS channel number of the first DMA transfer error after the DTSER bit is cleared to 0. These bits are read-only and cannot be cleared. |
| 7 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DTSERWR | DTS DMA transfer error occurring cycle This bit is updated at the same time as setting of the DTS DMA transfer error flag (DTSER), indicating which cycle of read or write the DMA transfer error occurs in. This bit is not updated when a new DMA transfer error occurs after the DTSER bit has been set. If the DTSER bit is cleared, this bit is also cleared to 0. 0: DMA transfer error occurs in read cycle. 1: DMA transfer error occurs in write cycle. |
| 0 | DTSER | DTS DMA transfer error flag This bit shows whether DMA transfer error is generated in the DTS. 0: DMA transfer error is not generated 1: DMA transfer error is generated If DMA transfer error is generated in the DTS while this bit is 0, this bit is set, and DTSERCH6-0 retains the DTS channel number of the DMA transfer error. If DMA transfer error is generated in the DTS while this bit is 1, this bit remains 1, and DTSERCH6-0 does not change. This bit can be cleared by using the DTSERC register. |

7.8.2.7 DTSER2 — DTS Error Register 2

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0028_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|--------------|----|----|----------------|----------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | RAM DED | RAMDE DOV | — | — | RAMDE DAD11 | RAMDE DAD10 | RAMDE DAD9 | RAMDE DAD8 | RAMDE DAD7 | RAMDE DAD6 | RAMDE DAD5 | RAMDE DAD4 | RAMDE DAD3 | RAMDE DAD2 | RAMDE DAD1 | RAMDE DAD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RAM SED | RAMSE DOV | — | — | RAMSE DAD11 | RAMSE DAD10 | RAMSE DAD9 | RAMSE DAD8 | RAMSE DAD7 | RAMSE DAD6 | RAMSE DAD5 | RAMSE DAD4 | RAMSE DAD3 | RAMSE DAD2 | RAMSE DAD1 | RAMSE DAD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.22 DTSER2 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------------|--|
| 31 | RAMDED | DTSRAM DED error flag This bit shows whether the DED error is generated in the read access to the DTSRAM. 0: DED error is not generated in the DTSRAM 1: DED error is generated in the DTSRAM If DED error is generated in the DTSRAM while this bit is 0, this bit is set, and RAMDEDAD11-0 retains the DTSRAM address of the error. If DED error is generated in the DTSRAM while this bit is 1, this bit remains 1, and RAMDEDAD11-0 does not change. This bit can be cleared by using the DTSERC register. |
| 30 | RAMDEDOV | DTSRAM DED error overflow flag This bit is set when the RAMDED bit is 1 and the DED error occurs in DTSRAM read access whose address is different from that specified by the RAMDEDAD11-0 bit. This bit can be cleared by operation of the DTSERC register. |
| 29, 28 | Reserved | When read, the value after reset is read. |
| 27 to 16 | RAMDEDAD [11:0] | DTSRAM DED error address These bits show the DTSRAM address of the first DTSRAM DED error after the RAMDED bit is cleared to 0. These bits are read-only and cannot be cleared. |
| 15 | RAMSED | DTSRAM SED error flag This bit shows whether the SED error is generated in the read access to the DTSRAM. 0: SED error is not generated in the DTSRAM 1: SED error is generated in the DTSRAM If SED error is generated in the DTSRAM while this bit is 0, this bit is set, and RAMSEDAD11-0 retains the DTSRAM address of the error. If SED error is generated in the DTSRAM while this bit is 1, this bit remains 1, and RAMSEDAD11-0 does not change. This bit can be cleared by using the DTSERC register. |
| 14 | RAMSEDOV | DTSRAM SED error overflow flag This bit is set when the RAMSED bit is 1 and the SED error occurs in DTSRAM read access whose address is different from that specified by the RAMSEDAD11-0 bit. This bit can be cleared by operation of the DTSERC register. |
| 13, 12 | Reserved | When read, the value after reset is read. |

Table 7.22 DTSER2 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 11 to 0 | RAMSEDAD [11:0] | DTSRAM SED error address These bits show the DTSRAM address of the first DTSRAM SED error after the RAMSED bit is cleared to 0. These bits are read-only and cannot be cleared. |

7.8.2.8 DTSERC — DTS Error Clear Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 002C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | RAMDEDC | RAMDEDOVC | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RAMSEDC | RAMSEDOVC | — | — | — | — | — | — | — | — | — | — | — | — | — | DTSERC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.23 DTSERC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 | RAMDEDC | DTSRAM DED error flag clear If a user writes 1 to this bit, the DTSRAM DED error flag (DTSER2.RAMDED) is cleared. The read value is always 0. |
| 30 | RAMDEDOVC | DTSRAM DED error overflow flag clear When the user writes 1 to this bit, the DTSRAM DED error overflow flag (DTSER2.RAMDEDOV) is cleared. The read value is always 0. |
| 29 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 | RAMSEDC | DTSRAM SED error flag clear If a user writes 1 to this bit, the DTSRAM SED error flag (DTSER2.RAMSED) is cleared. The read value is always 0. |
| 14 | RAMSEDOVC | DTSRAM SED error overflow flag clear When the user writes 1 to this bit, the DTSRAM SED error overflow flag (DTSER2.RAMSEDOV) is cleared. The read value is always 0. |
| 13 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | DTSERC | DTS error flag clear If a user writes 1 to this bit, the DTS DMA error flag (DTSER1.DTSER) is cleared. The read value is always 0. |

7.8.2.9 DM0CMV — DMAC0 Register Access Protection Violation Register

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0030_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|-------|----|----|----|----|----|----|-------|-------|-------|-------|-------|----|----|
| | PEID2 | PEID1 | PEID0 | — | — | — | — | — | — | SPID4 | SPID3 | SPID2 | SPID1 | SPID0 | UM | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | VCH2 | VCH1 | VCH0 | — | — | — | VF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.24 DM0CMV Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 29 | PEID[2:0] | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 28 to 23 | Reserved | When read, the value after reset is read. |
| 22 to 18 | SPID[4:0] | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 17 | UM | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 16 to 7 | Reserved | When read, the value after reset is read. |
| 6 to 4 | VCH[2:0] | Illegal access channel These bits show the channel number (0-7) of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | VF | Illegal access flag This bit shows whether illegal access occurs in the DMAC0. 0: No illegal access has occurred in the DMAC0 1: Illegal access has occurred in the DMAC0 If illegal access occurs in the DMAC0 while this bit is 0, this bit is set, and PEID, SPID, UM and VCH store their respective information. If illegal access occurs in the DMAC0 while this bit is 1, this bit remains 1, and PEID, SPID, UM and VCH do not change. This bit can be cleared by using the CMVC register. |

7.8.2.10 DM1CMV — DMAC1 Register Access Protection Violation Register

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0034_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|-------|----|----|----|----|----|----|-------|-------|-------|-------|-------|----|----|
| | PEID2 | PEID1 | PEID0 | — | — | — | — | — | — | SPID4 | SPID3 | SPID2 | SPID1 | SPID0 | UM | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | VCH2 | VCH1 | VCH0 | — | — | — | VF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.25 DM1CMV Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 29 | PEID[2:0] | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 28 to 23 | Reserved | When read, the value after reset is read. |
| 22 to 18 | SPID[4:0] | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 17 | UM | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 16 to 7 | Reserved | When read, the value after reset is read. |
| 6 to 4 | VCH[2:0] | Illegal access channel These bits show the channel number (0-7) of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | VF | Illegal access flag This bit shows whether illegal access occurs in the DMAC1. 0: No illegal access has occurred in the DMAC1 1: Illegal access has occurred in the DMAC1 If illegal access occurs in the DMAC1 while this bit is 0, this bit is set, and PEID, SPID, UM and VCH store their respective information. If illegal access occurs in the DMAC1 while this bit is 1, this bit remains 1, and PEID, SPID, UM and VCH do not change. This bit can be cleared by using the CMVC register. |

7.8.2.11 DTSCMV — DTS Register Access Protection Violation Register

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0038_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|-------|----|----|------|------|------|------|-------|-------|-------|-------|-------|----|----|
| | PEID2 | PEID1 | PEID0 | — | — | — | — | — | — | SPID4 | SPID3 | SPID2 | SPID1 | SPID0 | UM | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | VCH6 | VCH5 | VCH4 | VCH3 | VCH2 | VCH1 | VCH0 | — | — | — | VF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.26 DTSCMV Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 29 | PEID[2:0] | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 28 to 23 | Reserved | When read, the value after reset is read. |
| 22 to 18 | SPID[4:0] | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 17 | UM | Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 16 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 4 | VCH[6:0] | Illegal access channel These bits show the channel number (0-127) of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | VF | Illegal access flag This bit shows whether illegal access occurs in the DTS. 0: No illegal access has occurred in the DTS 1: Illegal access has occurred in the DTS If illegal access occurs in the DTS while this bit is 0, this bit is set, and PEID[2:0], SPID[4:0], UM and VCH[6:0] store their respective information. If illegal access occurs in the DTS while this bit is 1, this bit remains 1, and PEID[2:0], SPID[4:0], UM and VCH[6:0] do not change. This bit can be cleared by using the CMVC register. |

7.8.2.12 CMVC — Register Access Protection Violation Clear Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 003C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | DTSVC | DM1VC | DM0VC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 7.27 CMVC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | DTSVC | DTS illegal access flag clear The DTS illegal access flag (DTSCMV.VF) can be cleared by writing 1 to this bit. The read value is always 0. |
| 1 | DM1VC | DMAC1 illegal access flag clear The DMAC1 illegal access flag (DM1CMV.VF) can be cleared by writing 1 to this bit. The read value is always 0. |
| 0 | DM0VC | DMAC0 illegal access flag clear The DMAC0 illegal access flag (DM0CMV.VF) can be cleared by writing 1 to this bit. The read value is always 0. |

7.8.2.13 Transfer Status Register (TFRSTS)

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 004C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DTS CYC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DMA CY17 | DMA CY16 | DMA CY15 | DMA CY14 | DMA CY13 | DMA CY12 | DMA CY11 | DMA CY10 | DMA CY07 | DMA CY06 | DMA CY05 | DMA CY04 | DMA CY03 | DMA CY02 | DMA CY01 | DMA CY00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.28 TFRSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 17 | Reserved | When read, the value after reset is read. |
| 16 | DTSCYC | DMA cycle state of DTS IP block: 0: No DTS-DMA cycle executing 1: DTS-DMA cycle is executing After suspending DTS transfers, DTSCYC may be polled to assure that any possibly ongoing DTS cycle has finished. This flag is a copy of DTSSTS.DTSCYC. |
| 15 to 8 | DMACY[17:10] | DMA cycle execution state This bit shows whether a DMA cycle is ongoing in this channel. 0: DMA cycle is not ongoing. 1: DMA cycle is ongoing. After suspending a DMA channel, CY may be polled to assure that any possibly ongoing DMA cycle has finished. These flags are copies from DCSTn.CY. |
| 7 to 0 | DMACY[07:00] | DMA cycle execution state This bit shows whether a DMA cycle is ongoing in this channel. 0: DMA cycle is not ongoing. 1: DMA cycle is ongoing. After suspending a DMA channel, CY may be polled to assure that any possibly ongoing DMA cycle has finished. These flags are copies from DCSTn.CY. |

7.8.2.14 DTSPR_y — DTS Channel Priority Setting (y = 0 to 7)

- DTSPR₀

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0060_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|
| | DTS15 PR1 | DTS15 PR0 | DTS14 PR1 | DTS14 PR0 | DTS13 PR1 | DTS13 PR0 | DTS12 PR1 | DTS12 PR0 | DTS11 PR1 | DTS11 PR0 | DTS10 PR1 | DTS10 PR0 | DTS9 PR1 | DTS9 PR0 | DTS8 PR1 | DTS8 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS7 PR1 | DTS7 PR0 | DTS6 PR1 | DTS6 PR0 | DTS5 PR1 | DTS5 PR0 | DTS4 PR1 | DTS4 PR0 | DTS3 PR1 | DTS3 PR0 | DTS2 PR1 | DTS2 PR0 | DTS1 PR1 | DTS1 PR0 | DTS0 PR1 | DTS0 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.29 DTSPR₀ Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|---|
| 31 to 0 | DTS[15:0] PR[1:0] | DTS channel [15:0] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

- DTSPR₁

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0064_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | DTS31 PR1 | DTS31 PR0 | DTS30 PR1 | DTS30 PR0 | DTS29 PR1 | DTS29 PR0 | DTS28 PR1 | DTS28 PR0 | DTS27 PR1 | DTS27 PR0 | DTS26 PR1 | DTS26 PR0 | DTS25 PR1 | DTS25 PR0 | DTS24 PR1 | DTS24 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS23 PR1 | DTS23 PR0 | DTS22 PR1 | DTS22 PR0 | DTS21 PR1 | DTS21 PR0 | DTS20 PR1 | DTS20 PR0 | DTS19 PR1 | DTS19 PR0 | DTS18 PR1 | DTS18 PR0 | DTS17 PR1 | DTS17 PR0 | DTS16 PR1 | DTS16 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.30 DTSPR₁ Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|--|
| 31 to 0 | DTS[31:16] PR[1:0] | DTS channel [31:16] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

- DTSPR2

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0068_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | DTS47 PR1 | DTS47 PR0 | DTS46 PR1 | DTS46 PR0 | DTS45 PR1 | DTS45 PR0 | DTS44 PR1 | DTS44 PR0 | DTS43 PR1 | DTS43 PR0 | DTS42 PR1 | DTS42 PR0 | DTS41 PR1 | DTS41 PR0 | DTS40 PR1 | DTS40 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS39 PR1 | DTS39 PR0 | DTS38 PR1 | DTS38 PR0 | DTS37 PR1 | DTS37 PR0 | DTS36 PR1 | DTS36 PR0 | DTS35 PR1 | DTS35 PR0 | DTS34 PR1 | DTS34 PR0 | DTS33 PR1 | DTS33 PR0 | DTS32 PR1 | DTS32 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.31 DTSPR2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 31 to 0 | DTS[47:32] PR[1:0] | DTS channel [47:32] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

- DTSPR3

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 006C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | DTS63 PR1 | DTS63 PR0 | DTS62 PR1 | DTS62 PR0 | DTS61 PR1 | DTS61 PR0 | DTS60 PR1 | DTS60 PR0 | DTS59 PR1 | DTS59 PR0 | DTS58 PR1 | DTS58 PR0 | DTS57 PR1 | DTS57 PR0 | DTS56 PR1 | DTS56 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS55 PR1 | DTS55 PR0 | DTS54 PR1 | DTS54 PR0 | DTS53 PR1 | DTS53 PR0 | DTS52 PR1 | DTS52 PR0 | DTS51 PR1 | DTS51 PR0 | DTS50 PR1 | DTS50 PR0 | DTS49 PR1 | DTS49 PR0 | DTS48 PR1 | DTS48 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.32 DTSPR3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 31 to 0 | DTS[63:48] PR[1:0] | DTS channel [63:48] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

- DTSPR4

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0070_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | DTS79 PR1 | DTS79 PR0 | DTS78 PR1 | DTS78 PR0 | DTS77 PR1 | DTS77 PR0 | DTS76 PR1 | DTS76 PR0 | DTS75 PR1 | DTS75 PR0 | DTS74 PR1 | DTS74 PR0 | DTS73 PR1 | DTS73 PR0 | DTS72 PR1 | DTS72 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS71 PR1 | DTS71 PR0 | DTS70 PR1 | DTS70 PR0 | DTS69 PR1 | DTS69 PR0 | DTS68 PR1 | DTS68 PR0 | DTS67 PR1 | DTS67 PR0 | DTS66 PR1 | DTS66 PR0 | DTS65 PR1 | DTS65 PR0 | DTS64 PR1 | DTS64 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.33 DTSPR4 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 31 to 0 | DTS[79:64] PR[1:0] | DTS channel [79:64] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

- DTSPR5

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0074_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | DTS95 PR1 | DTS95 PR0 | DTS94 PR1 | DTS94 PR0 | DTS93 PR1 | DTS93 PR0 | DTS92 PR1 | DTS92 PR0 | DTS91 PR1 | DTS91 PR0 | DTS90 PR1 | DTS90 PR0 | DTS89 PR1 | DTS89 PR0 | DTS88 PR1 | DTS88 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS87 PR1 | DTS87 PR0 | DTS86 PR1 | DTS86 PR0 | DTS85 PR1 | DTS85 PR0 | DTS84 PR1 | DTS84 PR0 | DTS83 PR1 | DTS83 PR0 | DTS82 PR1 | DTS82 PR0 | DTS81 PR1 | DTS81 PR0 | DTS80 PR1 | DTS80 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.34 DTSPR5 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 31 to 0 | DTS[95:80] PR[1:0] | DTS channel [95:80] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

- DTSPR6

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0078_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | DTS111 PR1 | DTS111 PR0 | DTS110 PR1 | DTS110 PR0 | DTS109 PR1 | DTS109 PR0 | DTS108 PR1 | DTS108 PR0 | DTS107 PR1 | DTS107 PR0 | DTS106 PR1 | DTS106 PR0 | DTS105 PR1 | DTS105 PR0 | DTS104 PR1 | DTS104 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS103 PR1 | DTS103 PR0 | DTS102 PR1 | DTS102 PR0 | DTS101 PR1 | DTS101 PR0 | DTS100 PR1 | DTS100 PR0 | DTS99 PR1 | DTS99 PR0 | DTS98 PR1 | DTS98 PR0 | DTS97 PR1 | DTS97 PR0 | DTS96 PR1 | DTS96 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.35 DTSPR6 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 31 to 0 | DTS[111:96] PR[1:0] | DTS channel [111:96] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

- DTSPR7

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 007C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | DTS127 PR1 | DTS127 PR0 | DTS126 PR1 | DTS126 PR0 | DTS125 PR1 | DTS125 PR0 | DTS124 PR1 | DTS124 PR0 | DTS123 PR1 | DTS123 PR0 | DTS122 PR1 | DTS122 PR0 | DTS121 PR1 | DTS121 PR0 | DTS120 PR1 | DTS120 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DTS119 PR1 | DTS119 PR0 | DTS118 PR1 | DTS118 PR0 | DTS117 PR1 | DTS117 PR0 | DTS116 PR1 | DTS116 PR0 | DTS115 PR1 | DTS115 PR0 | DTS114 PR1 | DTS114 PR0 | DTS113 PR1 | DTS113 PR0 | DTS112 PR1 | DTS112 PR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.36 DTSPR7 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|--|
| 31 to 0 | DTS[127:112] PR[1:0] | DTS channel [127:112] priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest. |

7.8.2.15 DTRECCTL — DTSRAM ECC Control Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0080_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 7.37 DTRECCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31, 30 | PROT[1:0] | These bits enable or disable writing to the ECCDIS and SECDIS bits. The written data is not retained. The read value is always 0. These bits should be written when (PROT1, PROT0) = (0, 1). |
| 29 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | DTSRAM SED error correction disable This bit enables or disables SED error correction when the ECCDIS bit is 0. ECC 1 bit error detection operation is always executed when ECCDIS bit is 0 regardless of the state of this bit. Set (PROT1, PROT0) = (0, 1) at the same time as writing to this bit. 0: Error correction is enabled when the SED error is detected. 1: Error correction is disabled when the SED error is detected. |
| 0 | ECCDIS | DTSRAM ECC disable This bit enables or disables DTSRAM ECC error detection and correction. Set (PROT1, PROT0) = (0, 1) at the same time as writing to this bit. 0: Error detection and correction are enabled. 1: Error detection and correction are disabled. The encoding function is effective when error detection and correction are disabled. |

7.8.2.16 DTRERINT — DTSRAM Error Notification Control Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0084_H

Value after reset: 0000 0002_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 7.38 DTRERINT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | DTSRAM DED error external notification enable This bit enables or disables notification of DED error detection to the ECM when the ECCDIS bit in DTRECCTL is 0. 0: Notification of DED error to ECM is disabled. 1: Notification of DED error to ECM is enabled. |
| 0 | SEDIE | DTSRAM SED error external notification enable This bit enables or disables notification of SED error detection to the ECM when the ECCDIS bit in DTRECCTL is 0. 0: Notification of SED error to ECM is disabled. 1: Notification of SED error to ECM is enabled. |

7.8.2.17 DTRTCTL — DTSRAM Test Control Register

This register is used for ECC test (self-diagnosis). It enables setting of ECC test mode and selection of ECC data to be written to the DTSRAM.

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0094_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ECCTST | DATSEL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 7.39 DTRTCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31, 30 | PROT[1:0] | These bits enable or disable writing to the ECCTST and DATSEL bits. The written data is not retained. The read value is always 0. These bits should be written when (PROT1, PROT0) = (0, 1). |
| 29 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | ECCTST | DTSRAM ECC Test Mode This bit enables or disables DTSRAM ECC test mode. Set (PROT1, PROT0) = (0, 1) at the same time as writing to this bit. 0: ECC test mode is disabled. 1: ECC test mode is enabled. |
| 0 | DATSEL | ECC Test Data Selection This bit is valid when ECCTST is 1 and selects ECC data to be written to the DTSRAM. Set (PROT1, PROT0) = (0, 1) at the same time as writing to this bit. 0: ECC encoded from the written data is used. 1: The value specified by the DTSRAM test data writing register (DTRTWDAT) is used. |

7.8.2.18 DTRTWDAT — DTSRAM Test Write Data Register

This register is used for ECC test (self-diagnosis). It specifies ECC data to be written to the DTSRAM after ECC test mode is enabled (ECCTST = 1).

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0098_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|------------|------------|------------|------------|------------|------------|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | TWDAT 6 | TWDAT 5 | TWDAT 4 | TWDAT 3 | TWDAT 2 | TWDAT 1 | TWDAT 0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.40 DTRTWDAT Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | TWDAT[6:0] | ECC Test Write Data These bits specifies ECC data to be written to the DTSRAM when DTRTSCTL.ECCTST = 1 and DTRTSCTL.DATSEL = 1. Writing to this bit is enabled when DTRTSCTL.ECCTST = 1 When DTRTSCTL.ECCTST = 0, this bit cannot be written and its read value is 0. |

7.8.2.19 DTRTRDAT — DTSRAM Test Read Data Register

This register is used for ECC test (self-diagnosis). It reads out ECC data of the DTSRAM after ECC test mode is enabled (ECCTST = 1).

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 009C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | TRDAT6 | TRDAT5 | TRDAT4 | TRDAT3 | TRDAT2 | TRDAT1 | TRDAT0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.41 DTRTRDAT Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | TRDAT[6:0] | ECC Test Read Data This bits retains the last ECC data read out from the DTSRAM when DTRTSCTL.ECCTST = 1. When DTRTSCTL.ECCTST = 0, the read value of this bit is 0. |

7.8.2.20 ADECCTCL — ECC on BUS Address ECC Test Control Register

This register is used for ECC test (self-diagnosis). (refer **31.2.9.6, Test Function**)

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 00A0_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ECCTST | RWSEL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 7.42 ADECCTCL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31, 30 | PROT[1:0] | These bits enable or disable writing to the ECCTST and RWSEL bits. The written data is not retained. The read value is always 0. These bits should be written when (PROT1, PROT0) = (0, 1). |
| 29 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | ECCTST | Address ECC test mode Configures ECConBUS address ECC test mode. To update this bit, (PROT1, PROT0) = (0, 1) must be written at the same time. 0: Address ECC test mode is disabled. 1: Address ECC test mode is enabled. Address ECC output is replaced with the value of ADECCTDT register. |
| 0 | RWSEL | Address ECC test cycle selection Selects either read cycle or write cycle to test address ECC. This bit is valid only at ECCTST = 1. To update this bit, (PROT1, PROT0) = (0, 1) must be written at the same time. 0: Address ECC output is replaced with test data at DMA read cycle. 1: Address ECC output is replaced with test data at DMA write cycle. |

7.8.2.21 ADECCTDT — ECC on BUS Address ECC Test Data Register

This register is used for ECC test (self-diagnosis). (refer **31.2.9.6, Test Function**)

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 00A4_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | ECCDA T6 | ECCDA T5 | ECCDA T4 | ECCDA T3 | ECCDA T2 | ECCDA T1 | ECCDA T0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.43 ADECCTDT Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | ECCDAT[6:0] | ECC test data Specifies ECC test data used at ADECCTCL.ECCTST = 1. |

7.8.2.22 DMxCM — DMAC Channel Master Setting (x = 00 to 07, 10 to 17)

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0100_H + 4_H × Ch. No. n (n = 0 to 7)
<DMAC_base> + 0120_H + 4_H × Ch. No. n - 10 (n = 10 to 17)

Value after reset: 0000 2008_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|-------|----|----|----|----|----|----|-------|-------|-------|-------|-------|-----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PEID2 | PEID1 | PEID0 | — | — | — | — | — | — | SPID4 | SPID3 | SPID2 | SPID1 | SPID0 | UM | — |
| Value after reset | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 7.44 DMxCM Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 13 | PEID[2:0] | Channel master PEID setting Specifies the PEID information of the master assigned to the channel. |
| 12 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 to 7 | Reserved | When writing, always write 0. |
| 6 to 2 | SPID[4:0] | Channel master SPID setting Specifies the SPID information used by the master assigned to the channel. SPID = 0 and 1 are reserved and must not be used. If SPID = 0 or 1 is set, DMA will use SPID = 2. |
| 1 | UM | Channel master UM setting Specifies the UM information of the master assigned to the channel. |
| 0 | Reserved | When writing, always write 0. |

CAUTION

DM00CM to DM07CM configure the channel master information of the DMAC0 channel 0 to 7 respectively.

DM10CM to DM17CM configure the channel master information of the DMAC1 channel 0 to 7 respectively.

For information about the functions this register offers, see **7.6, Reliability Function**.

7.8.2.23 DTSmCM — DTS Channel Master Setting Register (m = 000 to 127)

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0200_H + 4_H × Ch. No. n (n = 0 to 127)

Value after reset: Undefined

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|-------|-------|-------|-------|-------|------|------|
| | PEID2 | PEID1 | PEID0 | — | — | — | — | — | — | SPID4 | SPID3 | SPID2 | SPID1 | SPID0 | UM | — |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W | R/W | R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CMC15 | CMC14 | CMC13 | CMC12 | CMC11 | CMC10 | CMC9 | CMC8 | CMC7 | CMC6 | CMC5 | CMC4 | CMC3 | CM2C | CMC1 | CMC0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.45 DTSmCM Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 29 | PEID[2:0] | Channel master PEID setting Specifies the PEID information of the master assigned to the channel. |
| 28 to 23 | Reserved | When writing, write the value “0”. When read, the unfixed value after reset and “0” after write is read. |
| 22 to 18 | SPID[4:0] | Channel master SPID setting Specifies the SPID information used by the master assigned to the channel. SPID = 0 and 1 are reserved and must not be used. If SPID = 0 or 1 is set, DMA will use SPID = 2. |
| 17 | UM | Channel master UM setting Specifies the UM information of the master assigned to the channel. |
| 16 | Reserved | When writing, write the value “0”. When read, the unfixed value after reset and “0” after write is read. |
| 15 to 0 | CMC[15:0] | Transfer count compare In terms of contents, this field is the same as the bit [15:0] of the 7.10.3.8, DTTCCm — DTS Transfer Count Compare Register . |

CAUTIONS

1. DTS000CM to DTS127CM configure the channel master information of the DTS channel 0 to 127 respectively.
2. Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

For information about the functions this register offers, see **7.6, Reliability Function**.

CAUTION

The lowest 16 bits of this register are shared with the DTS transfer count compare register, one of the DTS channel registers.

If you write to this register, the DTS transfer count compare register is updated as well.

Recommended setup procedure of the DTS channel master setting register

When the special master configures the overall DTS operation, the channel master setting must be configured in the bits 31 to 29, 22 to 16 in this register, and 0 must be specified in the bits 28 to 23, 15 to 0.

The bits 28 to 23 of this register are reserved, but you can read and write those bits.

7.9 DMAC Channel Register

7.9.1 DMAC Channel Register Address

Address = <DMAC_base> “FFFF 8000_H” + Offset address

Table 7.46 DMAC Channel Register Address

| Offset Address | Register Symbol | Meaning | Accessed Permission*1 | |
|--|-----------------|---------------------------------------|-----------------------|----------------|
| | | | Special Master | General Master |
| 0400 _H + 40 _H × [channel number] | DSAn | DMAC source address | √ | √ |
| 0404 _H + 40 _H × [channel number] | DDAn | DMAC destination address | √ | √ |
| 0408 _H + 40 _H × [channel number] | DTCn | DMAC transfer count | √ | √ |
| 040C _H + 40 _H × [channel number] | DTCTn | DMAC transfer control | √ | √ |
| 0410 _H + 40 _H × [channel number] | DRSAn | DMAC reload source address | √ | √ |
| 0414 _H + 40 _H × [channel number] | DRDAn | DMAC reload destination address | √ | √ |
| 0418 _H + 40 _H × [channel number] | DRTCn | DMAC reload transfer count | √ | √ |
| 041C _H + 40 _H × [channel number] | DTCCn | DMAC transfer count compare | √ | √ |
| 0420 _H + 40 _H × [channel number] | DCENn | DMAC channel operation enable setting | √ | √ |
| 0424 _H + 40 _H × [channel number] | DCSTn | DMAC transfer status | √ | √ |
| 0428 _H + 40 _H × [channel number] | DCSTSn | DMAC transfer status set | √ | √ |
| 042C _H + 40 _H × [channel number] | DCSTCn | DMAC transfer status clear | √ | √ |
| 0430 _H + 40 _H × [channel number] | DTFRn | DTFR setting | √ | √ |
| 0434 _H + 40 _H × [channel number] | DTFRRQn | DTFR transfer request status | √ | √ |
| 0438 _H + 40 _H × [channel number] | DTFRRQCn | DTFR transfer request clear | √ | √ |

Note: The [channel number] in the offset addresses and “n” in the register symbols are numbers in the range from 0 to 15, and the correspondence between the channel number n and the channel is as follows.

| Channel number n | Channel |
|------------------|-----------------|
| 0 | DMAC0 channel 0 |
| 1 | DMAC0 channel 1 |
| 2 | DMAC0 channel 2 |
| 3 | DMAC0 channel 3 |
| 4 | DMAC0 channel 4 |
| 5 | DMAC0 channel 5 |
| 6 | DMAC0 channel 6 |
| 7 | DMAC0 channel 7 |
| 8 | DMAC1 channel 0 |
| 9 | DMAC1 channel 1 |
| 10 | DMAC1 channel 2 |
| 11 | DMAC1 channel 3 |
| 12 | DMAC1 channel 4 |
| 13 | DMAC1 channel 5 |
| 14 | DMAC1 channel 6 |
| 15 | DMAC1 channel 7 |

Note 1. √: supported x: not supported

7.9.2 Details of DMAC Channel Registers

The “n” in the register symbols indicates the DMA channel number (n = 0 to 15).

7.9.2.1 DSA_n — DMAC Source Address Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0400_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | SA31 | SA30 | SA29 | SA28 | SA27 | SA26 | SA25 | SA24 | SA23 | SA22 | SA21 | SA20 | SA19 | SA18 | SA17 | SA16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | SA15 | SA14 | SA13 | SA12 | SA11 | SA10 | SA9 | SA8 | SA7 | SA6 | SA5 | SA4 | SA3 | SA2 | SA1 | SA0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.47 DSA_n Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 0 | SA[31:0] | Source address Specifies the DMA transfer source address. Those bits are updated whenever a DMA cycle is executed. If you read from those bits, the transfer source address for the next DMA cycle is read. |

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the source address is updated.
3. DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (x denotes an arbitrary one bit.)
The correct operation is not guaranteed if you select other setting.

| Data Size | SA3 | SA2 | SA1 | SA0 |
|-----------|-----|-----|-----|-----|
| 8 bits | x | x | x | x |
| 16 bits | x | x | x | 0 |
| 32 bits | x | x | 0 | 0 |
| 64 bits | x | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.9.2.2 DDAn — DMAC Destination Address Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0404_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | DA31 | DA30 | DA29 | DA28 | DA27 | DA26 | DA25 | DA24 | DA23 | DA22 | DA21 | DA20 | DA19 | DA18 | DA17 | DA16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | DA15 | DA14 | DA13 | DA12 | DA11 | DA10 | DA9 | DA8 | DA7 | DA6 | DA5 | DA4 | DA3 | DA2 | DA1 | DA0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.48 DDAn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | DA[31:0] | Destination address Specifies the DMA transfer destination address. Those bits are updated whenever a DMA cycle is executed. If you read from those bits, the transfer destination address for the next DMA cycle is read. |

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the destination address is updated.
3. DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.)
The correct operation is not guaranteed if you select other setting.

| Data Size | DA3 | DA2 | DA1 | DA0 |
|-----------|-----|-----|-----|-----|
| 8 bits | × | × | × | × |
| 16 bits | × | × | × | 0 |
| 32 bits | × | × | 0 | 0 |
| 64 bits | × | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.9.2.3 DTCn — DMAC Transfer Count Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0408_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| | ARC15 | ARC14 | ARC13 | ARC12 | ARC11 | ARC10 | ARC9 | ARC8 | ARC7 | ARC6 | ARC5 | ARC4 | ARC3 | ARC2 | ARC1 | ARC0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| | TRC15 | TRC14 | TRC13 | TRC12 | TRC11 | TRC10 | TRC9 | TRC8 | TRC7 | TRC6 | TRC5 | TRC4 | TRC3 | TRC2 | TRC1 | TRC0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.49 DTCn Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | |
|-------------------|--|---|-----------|-----------|-------------------|--|-------------------|--|---|---|-------------------|--|
| 31 to 16 | ARC[15:0] | <p>Address reload count</p> <p>Specifies the number of transfers until the address reload when the reload function 2 is used, and also specifies the number of transfers when the block transfer 2 is used. If you read from those bits during DMA transfer, the address reload count for the next DMA cycle is read.</p> <p>When the reload function 2 or block transfer 2 is used, ARC[15:0] is decremented by one for every DMA cycle. When the reload function 2 or block transfer 2 is not used, ARC[15:0] is not updated.</p> <p>If the value is 0000_H, it means that the number of transfers until the address reload when the reload function 2 is used and the number of transfers when the block transfer 2 is used are 65536.</p> | | | | | | | | | | |
| 15 to 0 | TRC[15:0] | <p>Transfer count</p> <p>Configures the number of transfers. TRC[15:0] is decremented by one whenever a DMA cycle is executed. If you read from those bits, the remaining number of transfers for the next DMA cycle is read. If the reload function is not used, after the last transfer is complete, the value at the completion (0000_H) is retained.</p> <table border="1"> <thead> <tr> <th>TRC[15:0]</th><th>Operation</th></tr> </thead> <tbody> <tr> <td>0000_H</td><td>The number of transfers is 65536, or the transfer is complete.</td></tr> <tr> <td>0001_H</td><td>The number of transfers or remaining transfers is 1.</td></tr> <tr> <td>:</td><td>:</td></tr> <tr> <td>FFFF_H</td><td>The number of transfers or remaining transfers is 65535.</td></tr> </tbody> </table> | TRC[15:0] | Operation | 0000 _H | The number of transfers is 65536, or the transfer is complete. | 0001 _H | The number of transfers or remaining transfers is 1. | : | : | FFFF _H | The number of transfers or remaining transfers is 65535. |
| TRC[15:0] | Operation | | | | | | | | | | | |
| 0000 _H | The number of transfers is 65536, or the transfer is complete. | | | | | | | | | | | |
| 0001 _H | The number of transfers or remaining transfers is 1. | | | | | | | | | | | |
| : | : | | | | | | | | | | | |
| FFFF _H | The number of transfers or remaining transfers is 65535. | | | | | | | | | | | |

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DCENn.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the transfer count and the address reload count are updated.

7.9.2.4 DTCTn — DMAC Transfer Control Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 040C_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|-----|-----|----|----|----|----|----|-------------|-------------|-------------|-------|-------|
| | — | — | — | — | ESE | DRS | — | — | — | — | — | CHNSE L2 | CHNSE L1 | SHNSE L0 | CHNE1 | CHNE0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|-----|-----|------------|------------|------------|------------|-------|-------|-------|-------|-----|-----|-----|------|------|
| | CCE | TCE | MLE | RLD2 M1 | RLD2 M0 | RLD1 M1 | RLD1 M0 | DACM1 | DACM0 | SACM1 | SACM0 | DS2 | DS1 | DS0 | TRM1 | TRM0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.50 DTCTn Register Contents (1/3)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 28 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 27 | ESE | Transfer error case DMA transfer disable setting Configures whether a DMA cycle is executed when the DCSTn.ER bit is set due to DMA transfer error. If this bit is cleared to 0, even when the DCSTn.ER bit is set due to DMA transfer error, the following DMA cycles can be executed. If this bit is set to 1, the following DMA cycles are not executed when the DCSTn.ER bit is set due to DMA transfer error. 0: DMA cycles are executed while the DCSTn.ER bit is set. 1: DMA cycles are not executed while the DCSTn.ER bit is set. |
| 26 | DRS | DMA transfer request selection assignment Selects the type of DMA transfer requests to be accepted. 0: Software DMA transfer request 1: Hardware DMA transfer request |
| 25 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 18 | CHNSEL[2:0] | Next channel in the chain Specifies the next channel in the chain. The next channel must be another channel in the same DMAC. You cannot specify a channel in the different DMAC or in the DTS. You cannot specify the same channel for the next channel. (If you do, the correct operation is not guaranteed.) |
| 17, 16 | CHNE1, 0 | Chain enable Selects the chain function. 00: Disabled 01: Chain at the last transfer A chain request is generated at the completion of the DMA cycle in which the remaining transfer count is one. 10: (Forbidden. No guarantee of operation) 11: Always chain A chain request is generated at the completion of every DMA cycle. |
| 15 | CCE | Transfer count match interrupt enable If this bit is set, a transfer count match interrupt is generated at the completion of the DMA cycle in which the transfer count compare register and the remaining transfer count have the same value. |

Table 7.50 DTCTn Register Contents (2/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|-----------|---|-------|-------|--------------------|---|---|-----------|---|---|-----------|---|---|-------|---|---|---------------------------------------|
| 14 | TCE | Transfer completion interrupt enable If this bit is set, a transfer completion interrupt is generated at the completion of the last transfer. A transfer completion notification interrupt is also generated at the completion of the last transfer when DCSTn.TC = 1. | | | | | | | | | | | | | | | |
| 13 | MLE | Continuous transfer enable If this bit is set, the DTE bit is not cleared at the completion of DMA transfer. In addition, even if the TC bit is not cleared, DMA transfer starts when there is a DMA transfer request. 0: The DTE bit is cleared at the completion of DMA transfer. In addition, the next DMA transfer can start only after the TC bit is cleared. 1: The DTE bit is not cleared at the completion of DMA transfer. In addition, even if the TC bit is not cleared, DMA transfer starts when there is a DMA transfer request. | | | | | | | | | | | | | | | |
| 12, 11 | RLD2M1, 0 | Reload function 2 setting Configures the reload function 2. 00: Reload function 2 is disabled. 01: Reload function 2 is enabled. The source address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 10: Reload function 2 is enabled. The destination address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 11: Reload function 2 is enabled. The source address, destination address, and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. | | | | | | | | | | | | | | | |
| 10, 9 | RLD1M1, 0 | Reload function 1 setting Configures the reload function 1. 00: Reload function 1 is disabled. 01: Reload function 1 is enabled. The source address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 10: Reload function 1 is enabled. The destination address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 11: Reload function 1 is enabled. The source address, destination address, and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) | | | | | | | | | | | | | | | |
| 8, 7 | DACM1, 0 | Destination address count direction Specifies the count direction of the destination address. <table border="1" data-bbox="678 1568 1417 1758"> <thead> <tr> <th>DACM1</th> <th>DACM0</th> <th>Direction of Count</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Increment</td> </tr> <tr> <td>0</td> <td>1</td> <td>Decrement</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fixed</td> </tr> <tr> <td>1</td> <td>1</td> <td>Forbidden (No guarantee of operation)</td> </tr> </tbody> </table> | DACM1 | DACM0 | Direction of Count | 0 | 0 | Increment | 0 | 1 | Decrement | 1 | 0 | Fixed | 1 | 1 | Forbidden (No guarantee of operation) |
| DACM1 | DACM0 | Direction of Count | | | | | | | | | | | | | | | |
| 0 | 0 | Increment | | | | | | | | | | | | | | | |
| 0 | 1 | Decrement | | | | | | | | | | | | | | | |
| 1 | 0 | Fixed | | | | | | | | | | | | | | | |
| 1 | 1 | Forbidden (No guarantee of operation) | | | | | | | | | | | | | | | |
| 6, 5 | SACM1, 0 | Source address count direction Specifies the count direction of the source address. <table border="1" data-bbox="678 1854 1417 2045"> <thead> <tr> <th>SACM1</th> <th>SACM0</th> <th>Direction of Count</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Increment</td> </tr> <tr> <td>0</td> <td>1</td> <td>Decrement</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fixed</td> </tr> <tr> <td>1</td> <td>1</td> <td>Forbidden (No guarantee of operation)</td> </tr> </tbody> </table> | SACM1 | SACM0 | Direction of Count | 0 | 0 | Increment | 0 | 1 | Decrement | 1 | 0 | Fixed | 1 | 1 | Forbidden (No guarantee of operation) |
| SACM1 | SACM0 | Direction of Count | | | | | | | | | | | | | | | |
| 0 | 0 | Increment | | | | | | | | | | | | | | | |
| 0 | 1 | Decrement | | | | | | | | | | | | | | | |
| 1 | 0 | Fixed | | | | | | | | | | | | | | | |
| 1 | 1 | Forbidden (No guarantee of operation) | | | | | | | | | | | | | | | |

Table 7.50 DTCTn Register Contents (3/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------|----------|--|---------------------------------------|-----|-----|--------------------|---|---|---|--------|---|---|---|---------|---|---|---|---------|---|---|---|---------|---|---|---|----------|----------------------|--|--|---------------------------------------|
| 4 to 2 | DS[2:0] | Transfer data size Specifies the transfer data size. | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>DS2</th> <th>DS1</th> <th>DS0</th> <th>Transfer Data Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>8 bits</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>16 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>32 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>64 bits</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>128 bits</td> </tr> <tr> <td colspan="3">Other than the above</td> <td>Forbidden (No guarantee of operation)</td> </tr> </tbody> </table> | DS2 | DS1 | DS0 | Transfer Data Size | 0 | 0 | 0 | 8 bits | 0 | 0 | 1 | 16 bits | 0 | 1 | 0 | 32 bits | 0 | 1 | 1 | 64 bits | 1 | 0 | 0 | 128 bits | Other than the above | | | Forbidden (No guarantee of operation) |
| DS2 | DS1 | DS0 | Transfer Data Size | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 8 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 16 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 32 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 64 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 128 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Other than the above | | | Forbidden (No guarantee of operation) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1, 0 | TRM1, 0 | Transfer mode Specifies the DMA transfer mode. 00: Single transfer 01: Block transfer 1 (The number of transfers is specified by the transfer count.) 10: Block transfer 2 (The number of transfers is specified by the address reload count.) 11: Forbidden (No guarantee of operation) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

CAUTIONS

1. Except for the case to clear DTCTn.MLE bit, it is forbidden to write to those bits when the channel operation is enabled (DCENn.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If forbidden settings are used for some of the bits, the correct operation is not guaranteed.

7.9.2.5 DRSA_n — DMAC Reload Source Address Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0410_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | RSA31 | RSA30 | RSA29 | RSA28 | RSA27 | RSA26 | RSA25 | RSA24 | RSA23 | RSA22 | RSA21 | RSA20 | RSA19 | RSA18 | RSA17 | RSA16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| | RSA15 | RSA14 | RSA13 | RSA12 | RSA11 | RSA10 | RSA9 | RSA8 | RSA7 | RSA6 | RSA5 | RSA4 | RSA3 | RSA2 | RSA1 | RSA0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.51 DRSA_n Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 0 | RSA[31:0] | Reload source address Specifies the source address to be reloaded to the DMA source address register when the reload function 1 and reload function 2 are used. |

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.) The correct operation is not guaranteed if you select other setting.

| Data Size | RSA3 | RSA2 | RSA1 | RSA0 |
|-----------|------|------|------|------|
| 8 bits | × | × | × | × |
| 16 bits | × | × | × | 0 |
| 32 bits | × | × | 0 | 0 |
| 64 bits | × | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.9.2.6 DRDAn — DMAC Reload Destination Address Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0414_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | RDA31 | RDA30 | RDA29 | RDA28 | RDA27 | RDA26 | RDA25 | RDA24 | RDA23 | RDA22 | RDA21 | RDA20 | RDA19 | RDA18 | RDA17 | RDA16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| | RDA15 | RDA14 | RDA13 | RDA12 | RDA11 | RDA10 | RDA9 | RDA8 | RDA7 | RDA6 | RDA5 | RDA4 | RDA3 | RDA2 | RDA1 | RDA0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.52 DRDAn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | RDA[31:0] | Reload destination address Specifies the destination address to be reloaded to the DMA destination address register when the reload function 1 or reload function 2 is used. |

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.) The correct operation is not guaranteed if you select other setting.

| Data Size | RDA3 | RDA2 | RDA1 | RDA0 |
|-----------|------|------|------|------|
| 8 bits | × | × | × | × |
| 16 bits | × | × | × | 0 |
| 32 bits | × | × | 0 | 0 |
| 64 bits | × | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.9.2.7 DRTCn — DMAC Reload Transfer Count Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0418_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | RARC 15 | RARC 14 | RARC 13 | RARC 12 | RARC 11 | RARC 10 | RARC9 | RARC8 | RARC7 | RARC6 | RARC5 | RARC4 | RARC3 | RARC2 | RARC1 | RARC0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RTRC 15 | RTRC 14 | RTRC 13 | RTRC 12 | RTRC 11 | RTRC 10 | RTRC9 | RTRC8 | RTRC7 | RTRC6 | RTRC5 | RTRC4 | RTRC3 | RTRC2 | RTRC1 | RTRC0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.53 DRTCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 16 | RARC[15:0] | Reload address reload count Specifies the value to be loaded to the address reload count in the transfer count register at the timing of reload when the reload function 2 is used. |
| 15 to 0 | RTRC[15:0] | Reload transfer count Specifies the value to be loaded to the transfer count in the transfer count register at the timing of reload when the reload function 1 (including a combination of reload function 1 and 2) is used. |

7.9.2.8 DTCCn — DMAC Transfer Count Compare Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 041C_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CMC15 | CMC14 | CMC13 | CMC12 | CMC11 | CMC10 | CMC9 | CMC8 | CMC7 | CMC6 | CMC5 | CMC4 | CMC3 | CMC2 | CMC1 | CMC0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.54 DTCCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | CMC[15:0] | Transfer count compare Configures the transfer count to be compared to the transfer count register. At the completion of the DMA cycle in which the remaining transfer count is the same as the value in this register, the transfer count match flag (DCSTn.CC) in the DMAC transfer status register is set. Furthermore, if the transfer count match interrupt enable (DTCTn.CCE) bit is 1, a transfer count match interrupt is generated. If 0000 _H is set, comparison with the transfer count is disabled. In this case, the transfer count match flag in the DMAC transfer status register is never set, and a transfer count match interrupt is never generated. |

CAUTION

It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1). If you do, the correct operation is not guaranteed.

7.9.2.9 DCENn — DMAC Channel Operation Enable Setting Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0420_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DTE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.55 DCENn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | DTE | <p>Channel operation enable</p> <p>Specifies whether to enable or disable the transfer operation of the channel. If the DTE bit is 1, DMA transfer starts when there is a DMA transfer request. If the MLE bit is 0, the DTE bit is cleared automatically at the completion of the DMA transfer. In addition, if 0 is written to the DTE bit during DMA transfer, the DMA transfer is suspended. If 1 is written to the DTE bit during suspension, the suspension is cleared and the DMA transfer resumes.</p> <p>0: Channel operation is disabled/Channel suspended 1: Channel operation is enabled/Channel suspension cleared</p> |

7.9.2.10 DCSTn — DMAC Transfer Status Register

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0424_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|------|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | ERWR | — | — | CY | ER | — | CC | TC | — | — | DR | SR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.56 DCSTn Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 12 | Reserved | When read, the value after reset is read. |
| 11 | ERWR | DMA Transfer Error occurring cycle This bit is updated at the same time as setting of the DMA transfer error flag (ER), indicating which cycle of read or write the DMA transfer error occurs in. This bit is not updated when a new DMA transfer error occurs after the ER bit has been set. If the ER bit is cleared, this bit is also cleared to 0. 0: DMA transfer error occurs in read cycle. 1: DMA transfer error occurs in write cycle. |
| 10, 9 | Reserved | When read, the value after reset is read. |
| 8 | CY | DMA cycle execution state This bit shows whether a DMA cycle is ongoing in this channel. 0: DMA cycle is not ongoing. 1: DMA cycle is ongoing. After suspending a DMA channel, CY may be polled to assure that any possibly ongoing DMA cycle has finished. |
| 7 | ER | Transfer error flag This bit is set when DMA transfer error is generated. If this bit is 1 and the DTCTn.ESE bit is set, a DMA cycle is not executed when a DMA transfer request is generated. 0: No DMA transfer error is generated 1: DMA transfer error is generated |
| 6 | Reserved | When read, the value after reset is read. |
| 5 | CC | Transfer count match flag This bit is set at the completion of the DMA cycle in which the remaining transfer count is the same as the value set in the transfer compare register. 0: No compare match has occurred with the transfer count compare register. 1: Compare match has occurred with the transfer count compare register. |
| 4 | TC | Transfer completion flag This bit is set at the completion of the last transfer and shows whether the DMA transfer is complete. If the MLE bit is 0 and this bit is 1, a DMA cycle is not executed when a DMA transfer request is generated. 0: DMA transfer incomplete 1: DMA transfer complete |
| 3, 2 | Reserved | When read, the value after reset is read. |

Table 7.56 DCSTn Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 1 | DR | <p>Hardware DMA transfer request status</p> <p>This bit shows whether there is a hardware DMA transfer request (DMARQ) from the DTFR.</p> <p>This bit changes regardless of the value of the DTE bit when a hardware DMA transfer request from the DTFR is generated. If the software DMA transfer request is selected by the transfer request selection bit (DTCTn.DRS) in the DMAC transfer control register, this bit is not set even when a hardware DMA transfer request is input from the DTFR.</p> <p>0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request</p> |
| 0 | SR | <p>Software DMA transfer request flag</p> <p>This bit shows whether there is a software DMA transfer request (DMARQ). This bit is automatically cleared at the completion of the last transfer. A user can set this bit by writing 1 to the DCSTS.SRS bit in the DMAC transfer status set register. In addition, a user can clear this bit by writing 1 to the SRC bit in the DMAC transfer status clear register, but if this is done, the ongoing DMA transfer is aborted and cannot be resumed.</p> <p>0: There is no software DMA transfer request 1: There is a software DMA transfer request</p> |

7.9.2.11 DCSTSn — DMAC Transfer Status Set Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0428_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SRS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.57 DCSTSn Set Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | SRS | Software DMA transfer request flag A user can set the software DMA transfer request flag (SR) by writing 1 to this bit. The read value is always 0. |

7.9.2.12 DCSTCn — DMAC Transfer Status Clear Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 042C_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-----|----|-----|-----|----|----|----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | ERC | — | CCC | TCC | — | — | — | SRC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R | R/W | R/W | R | R | R | R/W |

Table 7.58 DCSTCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 | ERC | Transfer error flag clear The DMA transfer error flag (ER) can be cleared by writing 1 to this bit. The read value is always 0. |
| 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | CCC | Transfer count match flag clear The transfer count match flag (CC) can be cleared by writing 1 to this bit. The read value is always 0. |
| 4 | TCC | Transfer completion flag clear The transfer completion flag (TC) can be cleared by writing 1 to this bit. The read value is always 0. |
| 3 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | SRC | Software DMA transfer request flag clear The software DMA transfer request flag (SR) can be cleared by writing 1 to this bit. The read value is always 0. |

7.9.2.13 DTFRn — DTFR Setting Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0430_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|---------|---------|---------|---------|---------|---------|---------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | REQSEL6 | REQSEL5 | REQSEL4 | REQSEL3 | REQSEL2 | REQSEL1 | REQSEL0 | REQEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.59 DTFRn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 to 1 | REQSEL[6:0] | Hardware DMA transfer source selection Selects one out of 128 hardware DMA transfer sources as the hardware DMA transfer request. 000_0000: Selecting the DMACTRG[0] input : 111_1111: Selecting the DMACTRG[127] input |
| 0 | REQEN | Hardware DMA transfer source selection enable This bit enables/disables the hardware DMA transfer source selection. 0: Hardware DMA transfer source selection is disabled. 1: Hardware DMA transfer source selection is enabled. If this bit is 0, even when the hardware DMA transfer source selected by the REQSEL6 to 0 bits is activated, it is not recognized as a hardware DMA transfer request, and a hardware DMA transfer request is not generated. |

CAUTIONS

1. When changing the DTFR.REQSEL bit, do so while DTFR.REQEN is 0.
2. If the hardware DMA transfer source selection enable bit (DTFRn.REQEN) is set to 0 while the DMAC is executing block transfer, transfer requests sent from the DTFR to the DMAC will be masked at the end of the currently executing DMA cycle and the block being transferred will be suspended.

7.9.2.14 DTFRRQn — DTFR Transfer Request Status Register

Access: This register can be read in 32-bit units.

Address: <DMAC_base> + 0434_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVF | DRQ |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.60 DTFRRQn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | OVF | DMA overflow flag If this bit is set, the DMA channel has received at least one more DMA requests than it can store or handle. One or more DMA requests are lost. OVF can only be read, not written. Write 1 to DTFRRQCn.OVFC to clear OVF. |
| 0 | DRQ | Hardware DMA transfer request status If this bit is set, it means that a hardware DMA transfer request exists or is retained. <ul style="list-style-type: none"> If the hardware DMA transfer request is an edge detection type*¹ This bit shows whether a hardware DMA transfer request generated by edge detection is retained. When the DMA transfer request acceptance signal (DMAAKn) from the DMAC is asserted, this bit is automatically cleared. A user can clear this bit by writing 1 to the DTFRRQCn.DRQC bit. If the hardware DMA transfer request is a level input type*¹ This bit shows whether there is a hardware DMA transfer request input from the outside. Even when the DMA transfer request acceptance signal (DMAAKn) from the DMAC is asserted, this bit is not automatically cleared. In addition, this bit is not cleared even when a user writes to the DTFRRQCn.DRQC bit. This bit changes regardless of the value of the DTFRn.REQEN bit when a hardware DMA transfer request from the outside is generated. 0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request |

Note 1. Whether the hardware DMA transfer request is an edge detection type or level input type depends on the hardware DMA transfer source selected by DTFRn.REQSEL. Note also that although the reception interrupts for the PSI50 and PSI51 DMA trigger sources (INTPSI50RI and INTPSI51RI) are level interrupts, a hardware DMA transfer request is acknowledged by the input of the edge detection signal to the DMAC.

7.9.2.15 DTFRRQCn — DTFR Transfer Request Clear Register

Access: This register can be read/written in 32-bit units.

Address: <DMAC_base> + 0438_H + 40_H × Ch. No. n (n = 0 to 15)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVFC | DRQC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 7.61 DTFRRQCn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | OVFC | DMA overflow flag clear Write 1 to OVFC to clear DTFRRQn.OVF. The read value is always 0. |
| 0 | DRQC | Hardware DMA transfer request clear If the hardware DMA transfer request is an edge detection type* ¹ , a user can clear the DTFRRQn.DRQ bit by writing 1 to this bit. If the hardware DMA transfer request is a level input type* ¹ , the DTFRRQn.DRQ bit cannot be cleared by writing to this bit. The read value is always 0. |

Note 1. Whether the hardware DMA transfer request is an edge detection type or level input type depends on the hardware DMA transfer source selected by DTFRn.REQSEL. The hardware DMA transfer request is only an edge detection type in this device.

7.10 DTS Channel Register

7.10.1 Transfer information of the DTS (TI)

7.10.1.1 Structure of the TI

The transfer information of the DTS is called TI. One set of TI consists of 32 bits. 8 sets of TI are assigned for each channel. The 8 sets of TI is called TI-A, TI-B, TI-C, TI-D, TI-E, TI-F, TI-G, and TI-H.

Figure 7.20 shows the structure of the TI.

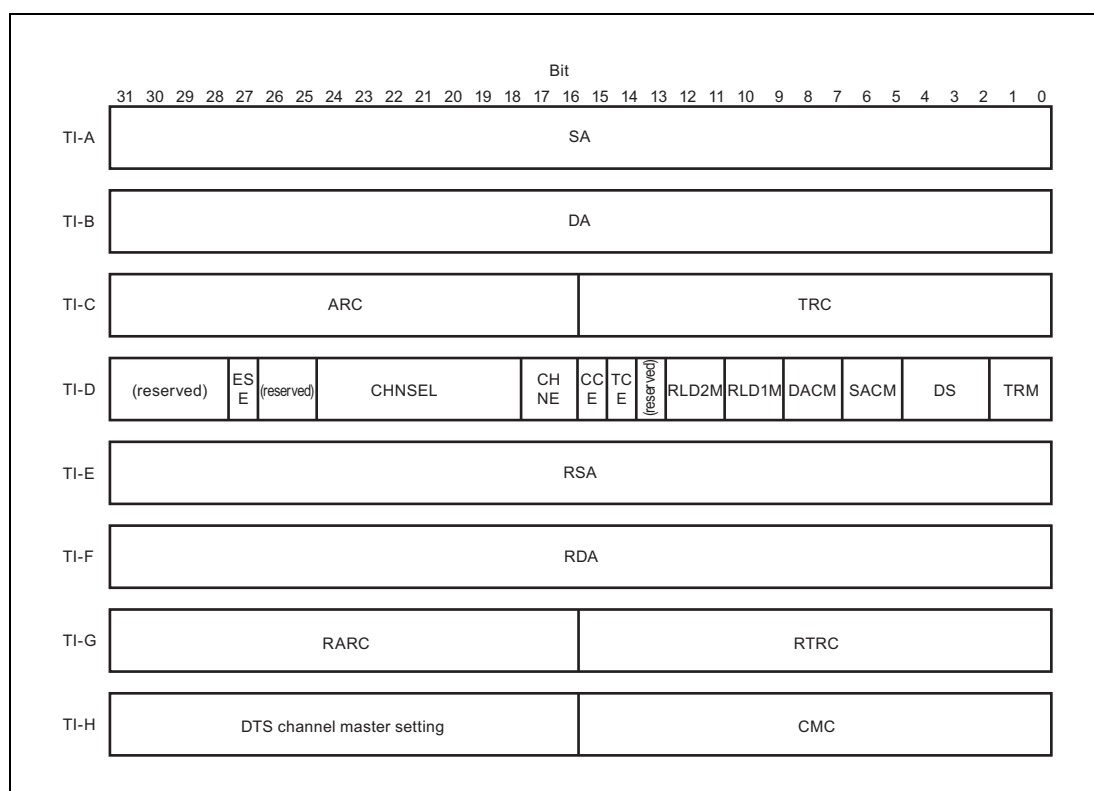


Figure 7.20 Structure of the TI

7.10.1.2 Organization of the TI in the DTSRAM

A user indirectly accesses the DTSRAM by way of the DTS channel registers for each channel and the DTS channel master setting registers.

Therefore, usually, a user does not have to think about the address organization of the TI in the DTSRAM.

As an exception, when ECC error occurs while the DTSRAM is read, the address of the ECC error in the DTSRAM is stored to the DTSRAM error register 2 (DTSER2), one of the global registers. In order to know which channel and TI have generated the error, you need to understand the address organization of the TI in the DTSRAM.

Figure 7.21 shows the address organization of the TI in the DTSRAM.

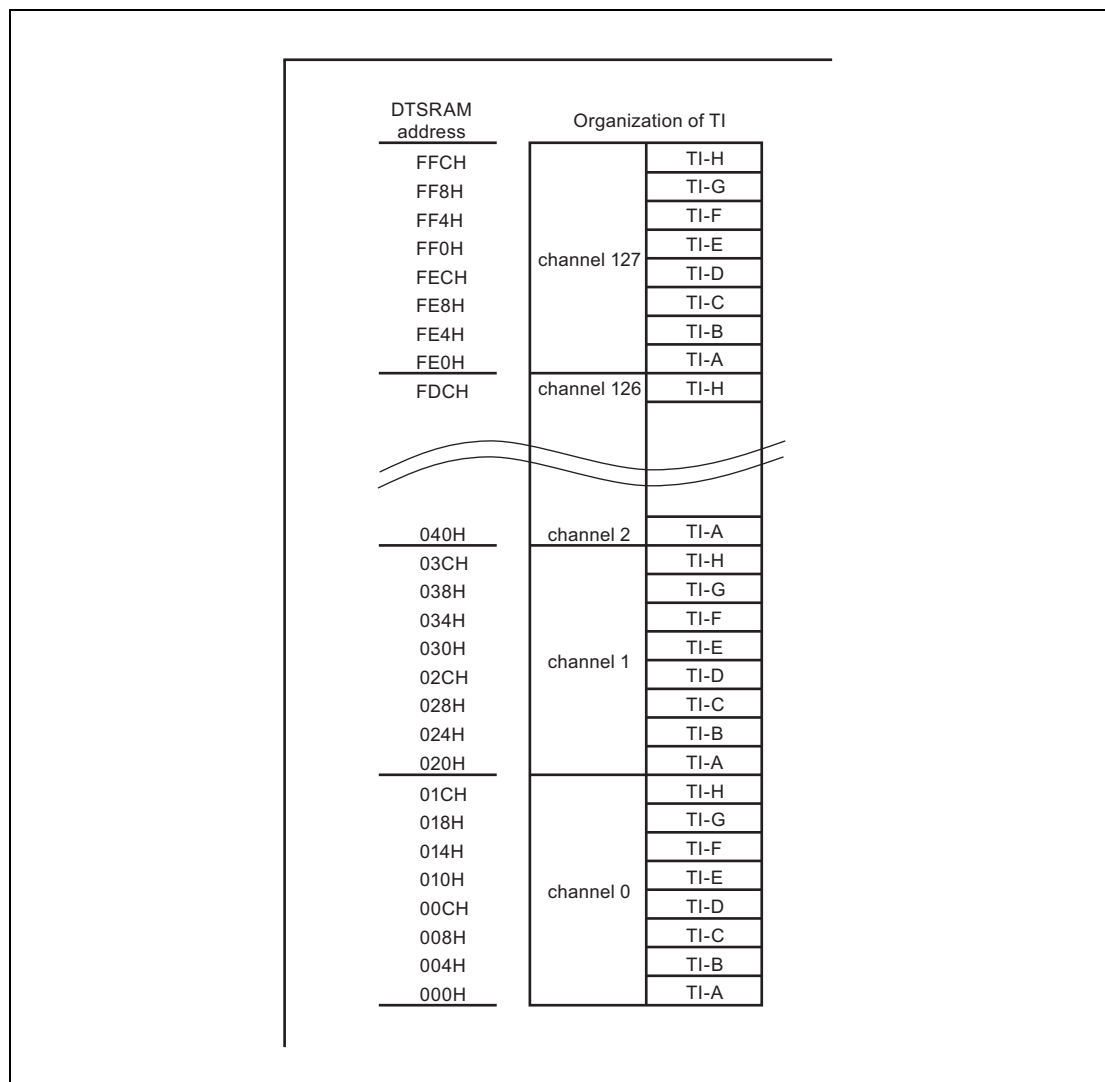


Figure 7.21 Organization of the TI in the DTSRAM

7.10.1.3 Accessing the TI

TI-A can be accessed by way of the DTS source address register (DTSAm) for each channel.

TI-B can be accessed by way of the DTS destination address register (DTDAm) for each channel.

TI-C can be accessed by way of the DTS transfer count register (DTTCm) for each channel.

TI-D can be accessed by way of the DTS transfer control register (DTTCTm) for each channel.

TI-E can be accessed by way of the DTS reload source address register (DTRSAm) for each channel.

TI-F can be accessed by way of the DTS reload destination address register (DTRDAm) for each channel.

TI-G can be accessed by way of the DTS reload transfer count register (DTRTCm) for each channel.

TI-H can be accessed by way of the channel master setting register (DTSmCM), which is a global register, and the transfer count compare register (DTTCCm) for each channel.

7.10.1.4 Caution about Accessing the TI

The data of the DTS channel master setting register and the data of the DTS transfer count compare register are stored to the same TI-H.

Access to the DTS channel master setting register (DTSmCM) is actually 32-bit access to the whole TI-H. Therefore, when you write to the DTS channel master setting register, the lower 16-bit data, which is the data for the DTS transfer count compare (CMC), is updated at the same time. When you read from the DTS channel master setting register, the value of the DTS transfer count compare (CMC) is read as the lower 16-bit data.

When you read from the DTS transfer count compare register (DTTCCm), 32-bit data is read from the TI-H, but only the lower 16-bit data is actually seen in the result of the register read. When you write to the DTS transfer count compare register (DTTCCm), lower 16-bit read/modify/write access to the 32-bit TI-H is used. Data in the TI immediately after the reset is undefined. It should be noted that, if you try to write to the DTS transfer count compare register (DTTCCm) before setting up the DTS channel master setting register, ECC error may be detected during the read of the read/modify/write access.

The bits 28 to 23 of the TI-H are not used, but you can read and write those bits by accessing the DTS channel master setting register. Our recommendation is as follows. When you write to those bits, write 0. When you read from those bits, discard the read value by software.

After the reset, the values of TI in DTSRAM are undefined. After the reset, if you read TI before you write to the TI, ECC error will occur.

Therefore, the first access to the following registers after the reset must be write access. The first access after the reset should never be read access.

- DTS source address register (DTSAm)
- DTS destination address register (DTDAm)
- DTS transfer count compare register (DTTCCm)
- DTS transfer control register (DTTCTm)
- DTS reload source address register (DTRSAm)
- DTS reload destination address register (DTRDAm)
- DTS reload transfer count register (DTRTCm)
- Channel master setting registers (DTSmCM)

In addition, the first access to DTS transfer count compare register (DTTCCm) after the reset must be done after write access to channel master setting register (DTSmCM).

You can access the TI from a CPU while the DTS is executing DMA transfer. But if you do so, the following should be noted.

- While a channel is executing DMA transfer, the TI of the channel should not be updated by TI access from CPU. If this situation happens, the result of the DMA transfer and the contents of the TI may mismatch.
- If TI access is requested from CPU while TI fetch or TI write back is executed, the TI access is executed after the completion of TI fetch or TI write back. If TI fetch or TI write back is requested while TI access request from CPU is processed, the TI fetch or TI write back is executed after the completion of TI access.

7.10.2 DTS Channel Register Address

Address = <DTS_base> “FFFF 9000_H” + Offset address

Table 7.62 DTS Channel Register Address

| Offset Address | Register Symbol | Meaning | Accessed Permission | |
|--|--------------------|--------------------------------|---------------------|----------------|
| | | | Special Master | General Master |
| 0000 _H + 40 _H × [channel number] | DTSA _m | DTS source address | enable | enable |
| 0004 _H + 40 _H × [channel number] | DTDA _m | DTS destination address | enable | enable |
| 0008 _H + 40 _H × [channel number] | DTTC _m | DTS transfer count | enable | enable |
| 000C _H + 40 _H × [channel number] | DTTCT _m | DTS transfer control | enable | enable |
| 0010 _H + 40 _H × [channel number] | DTRSA _m | DTS reload source address | enable | enable |
| 0014 _H + 40 _H × [channel number] | DTRDA _m | DTS reload destination address | enable | enable |
| 0018 _H + 40 _H × [channel number] | DTRTC _m | DTS reload transfer count | enable | enable |
| 001C _H + 40 _H × [channel number] | DTTCC _m | DTS transfer count compare | enable | enable |
| 0020 _H + 40 _H × [channel number] | DTFSL _m | DTSFSL operation setting | enable | enable |
| 0024 _H + 40 _H × [channel number] | DTFST _m | DTSFSL transfer request status | enable | enable |
| 0028 _H + 40 _H × [channel number] | DTFSS _m | DTSFSL transfer request set | enable | enable |
| 002C _H + 40 _H × [channel number] | DTFSC _m | DTSFSL transfer request clear | enable | enable |

Note 1. The [channel number] in the offset addresses is a number in the range from 0 to 127.
The “m” in the register symbols is a 3-digit number in the range from 000 to 127.

Note 2. Register access to the channels indicated as “reserved” in **Table 7.6, List of DTS Trigger Sources** is prohibited.

7.10.3 Details of DTS Channel Registers

The “m” in the register symbols indicates the DTS channel number (m = 000 to 127).

7.10.3.1 DTSAm — DTS Source Address Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0000_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | SA31 | SA30 | SA29 | SA28 | SA27 | SA26 | SA25 | SA24 | SA23 | SA22 | SA21 | SA20 | SA19 | SA18 | SA17 | SA16 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SA15 | SA14 | SA13 | SA12 | SA11 | SA10 | SA9 | SA8 | SA7 | SA6 | SA5 | SA4 | SA3 | SA2 | SA1 | SA0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.63 DTSAm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | SA[31:0] | Source address Specifies the DMA transfer source address. SA[31:0] is updated at the timing of the TI write back and retains the DMA transfer source address for the next DMA transfer. |

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (x denotes an arbitrary one bit.) The correct operation is not guaranteed if you select other setting.

| Data Size | SA3 | SA2 | SA1 | SA0 |
|-----------|-----|-----|-----|-----|
| 8 bits | x | x | x | x |
| 16 bits | x | x | x | 0 |
| 32 bits | x | x | 0 | 0 |
| 64 bits | x | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.10.3.2 DTDAm — DTS Destination Address Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0004_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | DA31 | DA30 | DA29 | DA28 | DA27 | DA26 | DA25 | DA24 | DA23 | DA22 | DA21 | DA20 | DA19 | DA18 | DA17 | DA16 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DA15 | DA14 | DA13 | DA12 | DA11 | DA10 | DA9 | DA8 | DA7 | DA6 | DA5 | DA4 | DA3 | DA2 | DA1 | DA0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.64 DTDAm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | DA[31:0] | Destination address Specifies the DMA transfer destination address. DA[31:0] is updated at the timing of the TI write back and retains the DMA transfer destination address for the next DMA transfer. |

CAUTIONS

1. If DMA transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the destination address is updated.
2. DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.) The correct operation is not guaranteed if you select other setting.

| Data Size | DA3 | DA2 | DA1 | DA0 |
|-----------|-----|-----|-----|-----|
| 8 bits | × | × | × | × |
| 16 bits | × | × | × | 0 |
| 32 bits | × | × | 0 | 0 |
| 64 bits | × | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.10.3.3 DTTCm — DTS Transfer Count Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0008_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | ARC15 | ARC14 | ARC13 | ARC12 | ARC11 | ARC10 | ARC9 | ARC8 | ARC7 | ARC6 | ARC5 | ARC4 | ARC3 | ARC2 | ARC1 | ARC0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TRC15 | TRC14 | TRC13 | TRC12 | TRC11 | TRC10 | TRC9 | TRC8 | TRC7 | TRC6 | TRC5 | TRC4 | TRC3 | TRC2 | TRC1 | TRC0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.65 DTTCm Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | |
|-------------------|--|---|-----------|-----------|-------------------|-----------------------------------|-------------------|--|---|---|-------------------|--|
| 31 to 16 | ARC[15:0] | <p>Address reload count</p> <p>Specifies the number of transfers until the address reload when the reload function 2 is used, and also specifies the number of transfers when the block transfer 2 is used. When the reload function 2 or block transfer 2 is used, ARC[15:0] is decremented by one for every DMA cycle and updated at the timing of the T1 write back. When the reload function 2 or block transfer 2 is not used, ARC[15:0] is not updated.</p> <p>If 0000_H is set, address reload is disabled.</p> <p>If the value is 0000_H at the start of the DMA cycle, the address reload count is not decremented even if a DMA cycle is generated.</p> | | | | | | | | | | |
| 15 to 0 | TRC[15:0] | <p>Transfer count</p> <p>Configures the number of transfers. TRC[15:0] is decremented by one whenever a DMA cycle is executed. It is updated at the timing of the T1 write back. If the reload function is not used, after the last transfer is complete, the value at the completion (0000_H) is retained.</p> <p>If 0000_H is set, DMA transfer is not executed even when a DMA transfer request is accepted.</p> <table border="1"> <thead> <tr> <th>TRC[15:0]</th> <th>Operation</th> </tr> </thead> <tbody> <tr> <td>0000_H</td> <td>Transfer is disabled or complete.</td> </tr> <tr> <td>0001_H</td> <td>The number of transfers or remaining transfers is 1.</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>FFFF_H</td> <td>The number of transfers or remaining transfers is 65535.</td> </tr> </tbody> </table> | TRC[15:0] | Operation | 0000 _H | Transfer is disabled or complete. | 0001 _H | The number of transfers or remaining transfers is 1. | : | : | FFFF _H | The number of transfers or remaining transfers is 65535. |
| TRC[15:0] | Operation | | | | | | | | | | | |
| 0000 _H | Transfer is disabled or complete. | | | | | | | | | | | |
| 0001 _H | The number of transfers or remaining transfers is 1. | | | | | | | | | | | |
| : | : | | | | | | | | | | | |
| FFFF _H | The number of transfers or remaining transfers is 65535. | | | | | | | | | | | |

CAUTIONS

1. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the transfer count and the address reload count are updated.
2. Unlike a DMAC, "0000_H" in the transfer count of the DTS does not mean "65536 transfers" but means that transfer is disabled or complete.

7.10.3.4 DTTCTm — DTS Transfer Control Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 000C_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----|-----|----|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|----------|-------|-------|
| | — | — | — | — | ESE | — | — | CHNSE L6 | CHNSE L5 | CHNSE L4 | CHNSE L3 | CHNSE L2 | CHNSE L1 | CHNSE L0 | CHNE1 | CHNE0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CCE | TCE | — | RLD2 M1 | RLD2 M0 | RLD1 M1 | RLD1 M0 | DACM1 | DACM0 | SACM1 | SACM0 | DS2 | DS1 | DS0 | TRM1 | TRM0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.66 DTTCTm Register Contents (1/3)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 28 | Reserved | When writing, write the value “0”. When read, the unfixed value after reset and “0” after write is read. |
| 27 | ESE | Transfer error case DMA transfer abort setting Specifies whether to abort DMA transfer when DMA transfer error is generated. If this bit is cleared to 0, DMA transfer continues when DMA transfer error is generated. If this bit is set to 1, DMA transfer is aborted when DMA transfer error is generated. 0: DMA transfer continues when DMA transfer error is generated. 1: DMA transfer is aborted when DMA transfer error is generated. |
| 26, 25 | Reserved | When writing, write the value “0”. When read, the unfixed value after reset and “0” after write is read. |
| 24 to 18 | CHNSE[6:0] | Next channel in the chain Specifies the next channel in the chain. The next channel must be another channel in the DTS. You cannot specify a channel in a DMAC. You cannot specify the same channel for the next channel. (If you do, the correct operation is not guaranteed.) |
| 17, 16 | CHNE1, 0 | Chain enable Selects the chain function. 00: Disabled 01: Chain at the last transfer A chain request is generated at the completion of the DMA cycle in which the remaining transfer count is one. 10: (Forbidden. No guarantee of operation) 11: Always chain A chain request is generated at the completion of every DMA cycle. |
| 15 | CCE | Transfer count match interrupt enable If this bit is set, a transfer count match interrupt is generated at the completion of the DMA cycle in which the transfer count compare register and the remaining transfer count have the same value. |
| 14 | TCE | Transfer completion interrupt enable If this bit is set, a transfer completion interrupt is generated at the completion of the last transfer. |
| 13 | Reserved | When writing, write the value “0”. When read, the unfixed value after reset and “0” after write is read. |

Table 7.66 DTTCTm Register Contents (2/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------|-----------|---|---------------------------------------|-------|--------------------|--------------------|---|-----------|---|--------|-----------|---|---|---------|---|---|---------------------------------------|---------|---|---|---|---------|---|---|---|----------|----------------------|--|--|---------------------------------------|
| 12, 11 | RLD2M1, 0 | <p>Reload function 2 setting Configures the reload function 2.</p> <p>00: Reload function 2 is disabled. 01: Reload function 2 is enabled. The source address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1.</p> <p>10: Reload function 2 is enabled. The destination address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1.</p> <p>11: Reload function 2 is enabled. The source address, destination address, and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10, 9 | RLD1M1, 0 | <p>Reload function 1 setting Configures the reload function 1.</p> <p>00: Reload function 1 is disabled. 01: Reload function 1 is enabled. The source address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.)</p> <p>10: Reload function 1 is enabled. The destination address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.)</p> <p>11: Reload function 1 is enabled. The source address, destination address, and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.)</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8, 7 | DACM1, 0 | <p>Destination address count direction Specifies the count direction of the destination address.</p> <table border="1"> <thead> <tr> <th>DACM1</th> <th>DACM0</th> <th>Direction of Count</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Increment</td> </tr> <tr> <td>0</td> <td>1</td> <td>Decrement</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fixed</td> </tr> <tr> <td>1</td> <td>1</td> <td>Forbidden (No guarantee of operation)</td> </tr> </tbody> </table> | DACM1 | DACM0 | Direction of Count | 0 | 0 | Increment | 0 | 1 | Decrement | 1 | 0 | Fixed | 1 | 1 | Forbidden (No guarantee of operation) | | | | | | | | | | | | | |
| DACM1 | DACM0 | Direction of Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Increment | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Decrement | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Fixed | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Forbidden (No guarantee of operation) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6, 5 | SACM1, 0 | <p>Source address count direction Specifies the count direction of the source address.</p> <table border="1"> <thead> <tr> <th>SACM1</th> <th>SACM0</th> <th>Direction of Count</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Increment</td> </tr> <tr> <td>0</td> <td>1</td> <td>Decrement</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fixed</td> </tr> <tr> <td>0</td> <td>1</td> <td>Forbidden (No guarantee of operation)</td> </tr> </tbody> </table> | SACM1 | SACM0 | Direction of Count | 0 | 0 | Increment | 0 | 1 | Decrement | 1 | 0 | Fixed | 0 | 1 | Forbidden (No guarantee of operation) | | | | | | | | | | | | | |
| SACM1 | SACM0 | Direction of Count | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Increment | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Decrement | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Fixed | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Forbidden (No guarantee of operation) | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 to 2 | DS[2:0] | <p>Transfer data size Specifies the transfer data size.</p> <table border="1"> <thead> <tr> <th>DS2</th> <th>DS1</th> <th>DS0</th> <th>Transfer Data Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>8 bits</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>16 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>32 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>64 bits</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>128 bits</td> </tr> <tr> <td colspan="3">Other than the above</td> <td>Forbidden (No guarantee of operation)</td> </tr> </tbody> </table> | DS2 | DS1 | DS0 | Transfer Data Size | 0 | 0 | 0 | 8 bits | 0 | 0 | 1 | 16 bits | 0 | 1 | 0 | 32 bits | 0 | 1 | 1 | 64 bits | 1 | 0 | 0 | 128 bits | Other than the above | | | Forbidden (No guarantee of operation) |
| DS2 | DS1 | DS0 | Transfer Data Size | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 8 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 16 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 32 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 64 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 128 bits | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Other than the above | | | Forbidden (No guarantee of operation) | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 7.66 DTTCTm Register Contents (3/3)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 1, 0 | TRM1, 0 | Transfer mode Specifies the DMA transfer mode. 00: Single transfer 01: Block transfer 1 (The number of transfers is specified by the transfer count.) 10: Block transfer 2 (The number of transfers is specified by the address reload count.) 11: Forbidden (No guarantee of operation) |

CAUTIONS

1. If forbidden settings are used for some of the bits, the correct operation is not guaranteed.
2. The bits 31-28, 26, 25, and bit 13 are reserved. Our recommendation is as follows. When you write to those bits, write 0. When you read from those bits, discard the read value.

7.10.3.5 DTRSAm — DTS Reload Source Address Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0010_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RSA31 | RSA30 | RSA29 | RSA28 | RSA27 | RSA26 | RSA25 | RSA24 | RSA23 | RSA22 | RSA21 | RSA20 | RSA19 | RSA18 | RSA17 | RSA16 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RSA15 | RSA14 | RSA13 | RSA12 | RSA11 | RSA10 | RSA9 | RSA8 | RSA7 | RSA6 | RSA5 | RSA4 | RSA3 | RSA2 | RSA1 | RSA0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.67 DTRSAm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | RSA[31:0] | Reload source address Specifies the source address to be reloaded when the reload function 1 or reload function 2 is used. |

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.) The correct operation is not guaranteed if you select other setting.

| Data Size | RSA3 | RSA2 | RSA1 | RSA0 |
|-----------|------|------|------|------|
| 8 bits | × | × | × | × |
| 16 bits | × | × | × | 0 |
| 32 bits | × | × | 0 | 0 |
| 64 bits | × | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.10.3.6 DTRDAm — DTS Reload Destination Address Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0014_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RDA31 | RDA30 | RDA29 | RDA28 | RDA27 | RDA26 | RDA25 | RDA24 | RDA23 | RDA22 | RDA21 | RDA20 | RDA19 | RDA18 | RDA17 | RDA16 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RDA15 | RDA14 | RDA13 | RDA12 | RDA11 | RDA10 | RDA9 | RDA8 | RDA7 | RDA6 | RDA5 | RDA4 | RDA3 | RDA2 | RDA1 | RDA0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.68 DTRDAm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | RDA[31:0] | Reload destination address Specifies the destination address to be reloaded when the reload function 1 or reload function 2 is used. |

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.) The correct operation is not guaranteed if you select other setting.

| Data Size | RDA3 | RDA2 | RDA1 | RDA0 |
|-----------|------|------|------|------|
| 8 bits | × | × | × | × |
| 16 bits | × | × | × | 0 |
| 32 bits | × | × | 0 | 0 |
| 64 bits | × | 0 | 0 | 0 |
| 128 bits | 0 | 0 | 0 | 0 |

7.10.3.7 DTRTCm — DTS Reload Transfer Count Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0018_H + 40_H × Ch. No. m (m = 0 to 127)

Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | RARC 15 | RARC 14 | RARC 13 | RARC 12 | RARC 11 | RARC 10 | RARC 9 | RARC 8 | RARC 7 | RARC 6 | RARC 5 | RARC 4 | RARC 3 | RARC 2 | RARC 1 | RARC 0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RTRC 15 | RTRC 14 | RTRC 13 | RTRC 12 | RTRC 11 | RTRC 10 | RTRC 9 | RTRC 8 | RTRC 7 | RTRC 6 | RTRC 5 | RTRC 4 | RTRC 3 | RTRC 2 | RTRC 1 | RTRC 0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.69 DTRTCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 31 to 16 | RARC[15:0] | Reload address reload count Specifies the value to be reloaded to the address reload count when the reload function 2 is used. |
| 15 to 0 | RTRC[15:0] | Reload transfer count Specifies the value to be reloaded to the transfer count when the reload function 1 or reload function 2 is used. |
| | RTRC[15:0] | Operation |
| | 0000 _H | No DMA transfer |
| | 0001 _H | 1 transfer |
| | : | : |
| | FFFF _H | 65535 transfers |

7.10.3.8 DTTCCm — DTS Transfer Count Compare Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 001C_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CMC15 | CMC14 | CMC13 | CMC12 | CMC11 | CMC10 | CMC9 | CMC8 | CMC7 | CMC6 | CMC5 | CMC4 | CMC3 | CMC2 | CMC1 | CMC0 |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.70 DTTCCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | CMC[15:0] | Transfer count compare Configures the transfer count to be compared to the transfer count register. If the transfer count match interrupt enable (DTTCTm.CCE) bit in the DTS transfer control register is 1, a transfer count match interrupt is generated at the completion of the DMA cycle in which the remaining transfer count is the same as the value in this register. If 0000 _H is set, comparison with the transfer count is disabled. In this case, a transfer count match interrupt is not generated. |

CAUTION

This register must be accessed after the DTS channel master setting register is set up.
If you access this register without setting up the DTS channel master setting register after the reset, ECC error may be generated during access to this register.

7.10.3.9 DTFSLm — DTSFSL Operation Setting Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0020_H + 40_H × Ch. No. m (m = 0 to 127)
Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | REQEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.71 DTFSLm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | REQEN | <p>DMA transfer request enable</p> <p>This bit selects whether a DMA transfer request from this channel retained in the DTSFSL is used as a candidate in DTS channel arbitration.</p> <p>0: A DMA transfer request from this channel is not used as a candidate in DTS channel arbitration.</p> <p>1: A DMA transfer request from this channel is used as a candidate in DTS channel arbitration.</p> <p>If this bit is 0, even if the DTSFSL retains a DMA transfer request, this channel is not a candidate in DTS channel arbitration inside the DTSFSL, and consequently a DMA transfer request from this channel is not generated.</p> |

7.10.3.10 DTFSTm — DTSFSL Transfer Request Status Register

Access: This register can be read in 32-bit units.

Address: <DTS_base> + 0024_H + 40_H × Ch. No. m (m = 0 to 127)

Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVF | DRQ |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 7.72 DTFSTm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | OVF | DTS overflow flag If this bit is set, the DTS channel has received at least one more DMA requests than it can store or handle. One or more DMA requests are lost. OVF can only be read, not written. Write 1 to DTFSCm.OVFC to clear OVF. |
| 0 | DRQ | DMA transfer request pending This bit shows whether a DMA transfer request of this channel is pending. This bit is set when a rising edge is detected in the hardware transfer source input from the DTSTRG pin, or when software writes "1" to the DTFSSm.DRQ bit. This bit is automatically cleared when the DMA transfer request acceptance signal (DTSACK) from the DTS is asserted while the DTSFSL is requesting DMA transfer of this channel. Alternatively, software can clear this bit by writing 1 to the DTFSCm.DRQC bit. 0: A DMA transfer request is not pending. 1: A DMA transfer request is pending. |

7.10.3.11 DTFSSm — DTSFSL Transfer Request Set Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 0028_H + 40_H × Ch. No. m (m = 0 to 127)
 Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DRQS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 7.73 DTFSSm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | DRQS | DMA transfer request set A user can set the DTFSTm.DRQ bit by writing 1 to this bit. The read value is always 0. |

7.10.3.12 DTFSCm — DTSFSL Transfer Request Clear Register

Access: This register can be read/written in 32-bit units.

Address: <DTS_base> + 002C_H + 40_H × Ch. No. m (m = 0 to 127)

Register access for channels that are shown as reserved in **Table 7.6, List of DTS Trigger Sources** is prohibited.

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVFC | DRQC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 7.74 DTFSCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | OVFC | DTS overflow flag clear Write 1 to OVFC to clear DTFSTm.OVF. The read value is always 0. |
| 0 | DRQC | DMA transfer request clear A user can clear the DTFSTm.DRQ bit by writing 1 to this bit. The read value is always 0. |

7.11 DMAC/DTS Trigger Select Registers

Address = <DMATRГ_base> “FFF9 9000_H” + Offset address

Table 7.75 DTS Trigger Select Registers Address

| Offset Address | Register Symbol | Meaning |
|-------------------|-----------------|--|
| 0000 _H | DMACTRGSEL0 | DMAC primary/secondary select register 0 |
| 0004 _H | DMACTRGSEL1 | DMAC primary/secondary select register 1 |
| 0008 _H | DTSTRGSEL0 | DTS primary/secondary select register 0 |
| 000C _H | DTSTRGSEL1 | DTS primary/secondary select register 1 |

7.11.1 DTS Trigger Select Register Descriptions

7.11.1.1 DMACTRGSEL0 — DMAC Primary/Secondary Select Register 0

Access: This register can be read/written in 32-bit units.

Address: <DMATRГ_base> + 0000_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | — | — | — | — | — | DMACS EL010 | DMACS EL009 | DMACS EL008 | DMACS EL007 | DMACS EL006 | DMACS EL005 | DMACS EL004 | DMACS EL003 | DMACS EL002 | DMACS EL001 | DMACS EL000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.76 DTSTRGSEL2 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | DMACSEL010 | DMACTRG[23] selector DMACSEL010 specifies the DMAC trigger input for DMAC channel 23 DMACSEL010 = 0: TAUD0 CH10 interrupt is selected. DMACSEL010 = 1: COM_FIFO_DMA_request_2 from RS-CANFD is selected. |
| 9 | DMACSEL009 | DMACTRG[22] selector DMACSEL009 specifies the DMAC trigger input for DMAC channel 22 DMACSEL009 = 0: TAUD0 CH9 interrupt is selected. DMACSEL009 = 1: COM_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 8 | DMACSEL008 | DMACTRG[21] selector DMACSEL008 specifies the DMAC trigger input for DMAC channel 21 DMACSEL008 = 0: TAUD0 CH8 interrupt is selected. DMACSEL008 = 1: COM_FIFO_DMA_request_0 from RS-CANFD is selected. |

Table 7.76 DTSTRGSEL2 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 7 | DMACSEL007 | DMACTRG[20] selector DMACSEL007 specifies the DMAC trigger input for DMAC channel 20 DMACSEL007 = 0: TAUD0 CH7 interrupt is selected. DMACSEL007 = 1: RX_FIFO_DMA_request_7 from RS-CANFD is selected. |
| 6 | DMACSEL006 | DMACTRG[19] selector DMACSEL006 specifies the DMAC trigger input for DMAC channel 19 DMACSEL006 = 0: TAUD0 CH6 interrupt is selected. DMACSEL006 = 1: RX_FIFO_DMA_request_6 from RS-CANFD is selected. |
| 5 | DMACSEL005 | DMACTRG[18] selector DMACSEL005 specifies the DMAC trigger input for DMAC channel 18 DMACSEL005 = 0: TAUD0 CH5 interrupt is selected. DMACSEL005 = 1: RX_FIFO_DMA_request_5 from RS-CANFD is selected. |
| 4 | DMACSEL004 | DMACTRG[17] selector DMACSEL004 specifies the DMAC trigger input for DMAC channel 17 DMACSEL004 = 0: TAUD0 CH4 interrupt is selected. DMACSEL004 = 1: RX_FIFO_DMA_request_4 from RS-CANFD is selected. |
| 3 | DMACSEL003 | DMACTRG[16] selector DMACSEL003 specifies the DMAC trigger input for DMAC channel 16 DMACSEL003 = 0: TAUD0 CH3 interrupt is selected. DMACSEL003 = 1: RX_FIFO_DMA_request_3 from RS-CANFD is selected. |
| 2 | DMACSEL002 | DMACTRG[15] selector DMACSEL002 specifies the DMAC trigger input for DMAC channel 15 DMACSEL002 = 0: TAUD0 CH2 interrupt is selected. DMACSEL002 = 1: RX_FIFO_DMA_request_2 from RS-CANFD is selected. |
| 1 | DMACSEL001 | DMACTRG[14] selector DMACSEL001 specifies the DMAC trigger input for DMAC channel 14 DMACSEL001 = 0: TAUD0 CH1 interrupt is selected. DMACSEL001 = 1: RX_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 0 | DMACSEL000 | DMACTRG[13] selector DMACSEL000 specifies the DMAC trigger input for DMAC channel 13 DMACSEL000 = 0: TAUD0 CH0 interrupt is selected. DMACSEL000 = 1: RX_FIFO_DMA_request_0 from RS-CANFD is selected. |

7.11.1.2 DMACTRGSEL1 — DMAC Primary/Secondary Select Register 1

Access: This register can be read/written in 32-bit units.

Address: <DMATRГ_base> + 0004_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | DMACS EL110 | DMACS EL109 | DMACS EL108 | DMACS EL107 | DMACS EL106 | DMACS EL105 | DMACS EL104 | DMACS EL103 | DMACS EL102 | DMACS EL101 | DMACS EL100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.77 DTSTRGSEL3 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | DMACSEL110 | DMACTRG[44] selector DMACSEL110 specifies the DMAC trigger input for DMAC channel 44 DMACSEL110 = 0: TAUD1 CH15 interrupt is selected. DMACSEL110 = 1: COM_FIFO_DMA_request_2 from RS-CANFD is selected. |
| 9 | DMACSEL109 | DMACTRG[43] selector DMACSEL109 specifies the DMAC trigger input for DMAC channel 43 DMACSEL109 = 0: TAUD1 CH14 interrupt is selected. DMACSEL109 = 1: COM_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 8 | DMACSEL108 | DMACTRG[42] selector DMACSEL108 specifies the DMAC trigger input for DMAC channel 42 DMACSEL108 = 0: TAUD1 CH13 interrupt is selected. DMACSEL108 = 1: COM_FIFO_DMA_request_0 from RS-CANFD is selected. |
| 7 | DMACSEL107 | DMACTRG[41] selector DMACSEL107 specifies the DMAC trigger input for DMAC channel 41 DMACSEL107 = 0: TAUD1 CH12 interrupt is selected. DMACSEL107 = 1: RX_FIFO_DMA_request_7 from RS-CANFD is selected. |
| 6 | DMACSEL106 | DMACTRG[40] selector DMACSEL106 specifies the DMAC trigger input for DMAC channel 40 DMACSEL106 = 0: TAUD1 CH11 interrupt is selected. DMACSEL106 = 1: RX_FIFO_DMA_request_6 from RS-CANFD is selected. |
| 5 | DMACSEL105 | DMACTRG[39] selector DMACSEL105 specifies the DMAC trigger input for DMAC channel 39 DMACSEL105 = 0: TAUD1 CH10 interrupt is selected. DMACSEL105 = 1: RX_FIFO_DMA_request_5 from RS-CANFD is selected. |
| 4 | DMACSEL104 | DMACTRG[38] selector DMACSEL104 specifies the DMAC trigger input for DMAC channel 38 DMACSEL104 = 0: TAUD1 CH9 interrupt is selected. DMACSEL104 = 1: RX_FIFO_DMA_request_4 from RS-CANFD is selected. |
| 3 | DMACSEL103 | DMACTRG[37] selector DMACSEL103 specifies the DMAC trigger input for DMAC channel 37 DMACSEL103 = 0: TAUD1 CH8 interrupt is selected. DMACSEL103 = 1: RX_FIFO_DMA_request_3 from RS-CANFD is selected. |

Table 7.77 DTSTRGSEL3 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 2 | DMACSEL102 | DMACTRG[36] selector DMACSEL102 specifies the DMAC trigger input for DMAC channel 36 DMACSEL102 = 0: TAUD1 CH7 interrupt is selected. DMACSEL102 = 1: RX_FIFO_DMA_request_2 from RS-CANFD is selected. |
| 1 | DMACSEL101 | DMACTRG[35] selector DMACSEL101 specifies the DMAC trigger input for DMAC channel 35 DMACSEL101 = 0: TAUD1 CH6 interrupt is selected. DMACSEL101 = 1: RX_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 0 | DMACSEL100 | DMACTRG[34] selector DMACSEL100 specifies the DMAC trigger input for DMAC channel 34 DMACSEL100 = 0: TAUD1 CH5 interrupt is selected. DMACSEL100 = 1: RX_FIFO_DMA_request_0 from RS-CANFD is selected. |

7.11.1.3 DTSTRGSEL0 — DTS Primary/Secondary Select Register 0

Access: This register can be read/written in 32-bit units.

Address: <DMATRГ_base> + 0008_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | DTSSEL010 | DTSSEL009 | DTSSEL008 | DTSSEL007 | DTSSEL006 | DTSSEL005 | DTSSEL004 | DTSSEL003 | DTSSEL002 | DTSSEL001 | DTSSEL000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.78 DTSTRGSEL0 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | DTSSEL010 | DTSTRG[23] selector DTSSEL010 specifies the DTS trigger input for DTS channel 23 DTSSEL010 = 0: TAUD0 CH10 interrupt is selected. DTSSEL010 = 1: COM_FIFO_DMA_request_2 from RS-CANFD is selected. |
| 9 | DTSSEL009 | DTSTRG[22] selector DTSSEL009 specifies the DTS trigger input for DTS channel 22 DTSSEL009 = 0: TAUD0 CH9 interrupt is selected. DTSSEL009 = 1: COM_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 8 | DTSSEL008 | DTSTRG[21] selector DTSSEL008 specifies the DTS trigger input for DTS channel 21 DTSSEL008 = 0: TAUD0 CH8 interrupt is selected. DTSSEL008 = 1: COM_FIFO_DMA_request_0 from RS-CANFD is selected. |
| 7 | DTSSEL007 | DTSTRG[20] selector DTSSEL007 specifies the DTS trigger input for DTS channel 20 DTSSEL007 = 0: TAUD0 CH7 interrupt is selected. DTSSEL007 = 1: RX_FIFO_DMA_request_7 from RS-CANFD is selected. |
| 6 | DTSSEL006 | DTSTRG[19] selector DTSSEL006 specifies the DTS trigger input for DTS channel 19 DTSSEL006 = 0: TAUD0 CH6 interrupt is selected. DTSSEL006 = 1: RX_FIFO_DMA_request_6 from RS-CANFD is selected. |
| 5 | DTSSEL005 | DTSTRG[18] selector DTSSEL005 specifies the DTS trigger input for DTS channel 18 DTSSEL005 = 0: TAUD0 CH5 interrupt is selected. DTSSEL005 = 1: RX_FIFO_DMA_request_5 from RS-CANFD is selected. |
| 4 | DTSSEL004 | DTSTRG[17] selector DTSSEL004 specifies the DTS trigger input for DTS channel 17 DTSSEL004 = 0: TAUD0 CH4 interrupt is selected. DTSSEL004 = 1: RX_FIFO_DMA_request_4 from RS-CANFD is selected. |
| 3 | DTSSEL003 | DTSTRG[16] selector DTSSEL003 specifies the DTS trigger input for DTS channel 16 DTSSEL003 = 0: TAUD0 CH3 interrupt is selected. DTSSEL003 = 1: RX_FIFO_DMA_request_3 from RS-CANFD is selected. |
| 2 | DTSSEL002 | DTSTRG[15] selector DTSSEL002 specifies the DTS trigger input for DTS channel 15 DTSSEL002 = 0: TAUD0 CH2 interrupt is selected. DTSSEL002 = 1: RX_FIFO_DMA_request_2 from RS-CANFD is selected. |

Table 7.78 DTSTRGSEL0 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 1 | DTSSEL001 | DTSTRG[14] selector DTSSEL001 specifies the DTS trigger input for DTS channel 14 DTSSEL001 = 0: TAUD0 CH1 interrupt is selected. DTSSEL001 = 1: RX_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 0 | DTSSEL000 | DTSTRG[13] selector DTSSEL000 specifies the DTS trigger input for DTS channel 13 DTSSEL000 = 0: TAUD0 CH0 interrupt is selected. DTSSEL000 = 1: RX_FIFO_DMA_request_0 from RS-CANFD is selected. |

7.11.1.4 DTSTRGSEL1 — DTS Primary/Secondary Select Register 1

Access: This register can be read/written in 32-bit units.

Address: <DMATRGR_base> + 000C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | DTSSEL110 | DTSSEL109 | DTSSEL108 | DTSSEL107 | DTSSEL106 | DTSSEL105 | DTSSEL104 | DTSSEL103 | DTSSEL102 | DTSSEL101 | DTSSEL100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 7.79 DTSTRGSEL1 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | DTSSEL110 | DTSTRG[44] selector DTSSEL110 specifies the DTS trigger input for DTS channel 44 DTSSEL110 = 0: TAUD1 CH15 interrupt is selected. DTSSEL110 = 1: COM_FIFO_DMA_request_2 from RS-CANFD is selected. |
| 9 | DTSSEL109 | DTSTRG[43] selector DTSSEL109 specifies the DTS trigger input for DTS channel 43 DTSSEL109 = 0: TAUD1 CH14 interrupt is selected. DTSSEL109 = 1: COM_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 8 | DTSSEL108 | DTSTRG[42] selector DTSSEL108 specifies the DTS trigger input for DTS channel 42 DTSSEL108 = 0: TAUD1 CH13 interrupt is selected. DTSSEL108 = 1: COM_FIFO_DMA_request_0 from RS-CANFD is selected. |
| 7 | DTSSEL107 | DTSTRG[41] selector DTSSEL107 specifies the DTS trigger input for DTS channel 41 DTSSEL107 = 0: TAUD1 CH12 interrupt is selected. DTSSEL107 = 1: RX_FIFO_DMA_request_7 from RS-CANFD is selected. |
| 6 | DTSSEL106 | DTSTRG[40] selector DTSSEL106 specifies the DTS trigger input for DTS channel 40 DTSSEL106 = 0: TAUD1 CH11 interrupt is selected. DTSSEL106 = 1: RX_FIFO_DMA_request_6 from RS-CANFD is selected. |
| 5 | DTSSEL105 | DTSTRG[39] selector DTSSEL105 specifies the DTS trigger input for DTS channel 39 DTSSEL105 = 0: TAUD1 CH10 interrupt is selected. DTSSEL105 = 1: RX_FIFO_DMA_request_5 from RS-CANFD is selected. |
| 4 | DTSSEL104 | DTSTRG[38] selector DTSSEL104 specifies the DTS trigger input for DTS channel 38 DTSSEL104 = 0: TAUD1 CH9 interrupt is selected. DTSSEL104 = 1: RX_FIFO_DMA_request_4 from RS-CANFD is selected. |
| 3 | DTSSEL103 | DTSTRG[37] selector DTSSEL103 specifies the DTS trigger input for DTS channel 37 DTSSEL103 = 0: TAUD1 CH8 interrupt is selected. DTSSEL103 = 1: RX_FIFO_DMA_request_3 from RS-CANFD is selected. |
| 2 | DTSSEL102 | DTSTRG[36] selector DTSSEL102 specifies the DTS trigger input for DTS channel 36 DTSSEL102 = 0: TAUD1 CH7 interrupt is selected. DTSSEL102 = 1: RX_FIFO_DMA_request_2 from RS-CANFD is selected. |

Table 7.79 DTSTRGSEL1 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 1 | DTSEL101 | DTSTRG[35] selector DTSEL101 specifies the DTS trigger input for DTS channel 35 DTSEL101 = 0: TAUD1 CH6 interrupt is selected. DTSEL101 = 1: RX_FIFO_DMA_request_1 from RS-CANFD is selected. |
| 0 | DTSEL100 | DTSTRG[34] selector DTSEL100 specifies the DTS trigger input for DTS channel 34 DTSEL100 = 0: TAUD1 CH5 interrupt is selected. DTSEL100 = 1: RX_FIFO_DMA_request_0 from RS-CANFD is selected. |

Section 8 Reset Controller

8.1 Features

The Reset Controller controls all factors that have an influence on the reset behavior of the device.

The device has several kinds of reset categories depending on the areas that are reset. Each of reset categories is triggered by one or multiple reset sources.

The relation between reset categories and their sources is shown in **Table 8.1**. The relation between reset categories and initialized area is shown in **Table 8.2**.

Table 8.1 Reset Categories and Reset Sources

| Reset Category | Source |
|---------------------|--|
| Power On Reset | <ul style="list-style-type: none"> Power On Reset*¹ Debug Initiated Reset |
| System Reset 1 | <ul style="list-style-type: none"> Pin Reset CVM Reset Debugger Disconnect Reset |
| System Reset 2 | <ul style="list-style-type: none"> Software Reset (by SWSRESA0) ECM Reset (if RESC0 = 0) |
| Application Reset 1 | <ul style="list-style-type: none"> Software Reset (by SWARESAS0) ECM Reset (if RESC0 = 1: initial setting) |

Note 1. Power On Reset is asserted when VCC supply voltage is under VPOC level and is released when VCC supply voltage is over VPOC level. About VPOC, please refer to **Section 9, Power Supply Circuit** and **Section 37, Electrical Specifications**.

Table 8.2 Reset Categories and Initialized Area (1/2)
(√: Reset (Initialized) —: Without Reset (Kept))

| Initialized Area (module) | | Reset category | | | |
|---------------------------|----------------------------------|----------------|----------------|----------------|---------------------|
| | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| PE1 | PE1 | √ | √ | √ | √ |
| | Local RAM | √ | √*2 | √*1 | √*1 |
| | Window Watchdog Timer 0 | √ | √ | √ | √ |
| RAM | Global RAM | √ | √ | √ | √*1 |
| RAM | DTS RAM | √ | √ | √ | √*1 |
| System Control | Operating Mode | — | √*4 | — | — |
| | Clock Controller | √ | √ | √ | √ |
| | Core Voltage Monitor | √ | √*5 | — | — |
| | Clock Monitor | √ | √ | √ | √ |
| IO | IO Buffer / IO Port Function | √ | √ | √ | √ |
| | $\overline{\text{ERROROUT}}$ Pin | √*3 | √*3 | √*3 | — |
| | $\overline{\text{RESETOUT}}$ Pin | √ | √ | √ | √ |
| | $\overline{\text{CVMOUT}}$ Pin | √ | — | — | — |

Table 8.2 Reset Categories and Initialized Area (2/2)
(√: Reset (Initialized) —: Without Reset (Kept))

| Initialized Area (module) | | Reset category | | | |
|---------------------------|--|----------------|----------------|----------------|---------------------|
| | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| Peripheral | BIST | √ | √ | √ | — |
| | Field BIST control register (BSEQ0CTL) | √ | √*6 | — | — |
| | ECM | √ | √ | √ | √ |
| | ECM Master/Checker Error Source Status Register 0/1/2 | √*7 | — | — | — |
| | ECM Error Output Clear Invalidation Configuration Register | √ | √ | √ | — |
| | CSIH | √ | √ | √ | √*1 |
| | Backup Register (BRAMDAT3-0) | — | — | — | — |
| | Other Peripherals | √ | √ | √ | √ |

Note 1. The execution of RAM initialization is configurable by a register setting.

Note 2. In case of Pin Reset, the execution of RAM initialization is configurable by a register setting.

Note 3. For details, please refer to **Section 32, Error Control Module (ECM)**.

Note 4. Pin Reset only.

Note 5. For details, please refer to **Section 10, Core Voltage Monitor**.

Note 6. CVM Reset only.

Note 7. Except Debug Initiated Reset.

8.2 Input/Output Pins

I/O pins related to reset are shown in **Table 8.3**.

Table 8.3 I/O Pins

| Pin function name | Direction | Description |
|-------------------|-----------|--------------|
| RESET | Input | Reset Input |
| RESETOUT | Output | Reset Output |

8.3 Register Description

The register list related to reset is shown in **Table 8.4**.

Registers can be protected from inadvertent write access due to erroneous program execution, etc. by configuration of the P-Bus Guards. For details, see **Section 31, Functional Safety**.

Table 8.4 List of Registers

| Address | Register Name | Description | Access Width | Value after Reset |
|------------------------|---------------|---|--------------|------------------------|
| FFF8 1000 _H | RESF | Reset Factor Register | 32 | 0000 0XXX _H |
| FFF8 1008 _H | RESFC | Reset Factor Clear Register | 32 | 0000 0000 _H |
| FFF8 1100 _H | SWSRESA0 | Software System Reset Request Register 0 | 32 | 0000 0000 _H |
| FFF8 1200 _H | SWARESAS0 | Software Application Reset Request Register 0 | 32 | 0000 0000 _H |
| FFF8 2800 _H | RESC | Reset Configuration Register | 32 | 0000 0001 _H |
| FFF8 1320 _H | STAC_DTSRAM | RAM Initialization Mode Control Register for DTS RAM | 32 | 0000 0003 _H |
| FFF8 1420 _H | STAC_GRAM | RAM Initialization Mode Control Register for Global RAM | 32 | 0000 0003 _H |
| FFF8 1520 _H | STAC_LM0 | RAM Initialization Mode Control Register for Local RAM | 32 | 0000 0003 _H |
| FFF8 1E20 _H | STAC_LM10 | RAM Initialization Mode Control Register for CSIH | 32 | 0000 0003 _H |

Register reset condition is shown in **Table 8.5**.

Table 8.5 Register Reset Condition

| Register Name | Reset Condition | | | |
|---------------|-----------------|----------------|----------------|---------------------|
| | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| RESF | √ | √*1 | | |
| RESFC | √ | √ | √ | √ |
| SWSRESA0 | √ | √ | √ | |
| SWARESAS0 | √ | √ | √ | |
| RESC | √ | √ | | |
| STAC_DTSRAM | √ | √ | √ | |
| STAC_GRAM | √ | √ | √ | |
| STAC_LM0 | √ | √*2 | | |
| STAC_LM10 | √ | √ | √ | |

Note 1. The register is initialized in case of CVM Reset only. Each bit of RESF is cleared as described in **Table 8.6**.

Note 2. The registers are initialized in case of CVM reset only.

8.3.1 RESF — Reset Factor Register

This register determines the reset source.

All bits are cleared by Power On Reset, Debugger Initiated Reset, CVM Reset, or software, except it is described differently in **Table 8.6**.

Access: This register can be read in 32-bit units.

Address: FFF8 1000_H

Value after reset: 0000 0XXX_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|------------|------------|----|------------|----|------------|----|------------|------------|------------|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | ARESF 3 | ARESF 2 | — | ARESF 0 | — | SRESF 4 | — | SRESF 2 | SRESF 1 | SRESF 0 | PRESF 0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | *1 | *1 | 0 | *1 | — | *1 | 0 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Note 1. The value differs with the reset factor.

Table 8.6 Reset Factor Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 11 | Reserved | When read, the value after reset is read. |
| 10 | ARESF3 | Filed BIST Execution Flag This flag is set to 1 after the Power On Reset, System Reset 1, or System Reset 2 if Field BIST is enabled. In this case the corresponding reset flag is asserted as well. This flag is only cleared by software. 0: Field BIST was not executed 1: Field BIST was executed |
| 9 | ARESF2 | ECM Application Reset Flag 0: No reset occurred 1: Reset has occurred |
| 8 | Reserved | When read, the value after reset is read. |
| 7 | ARESF0 | Software Application Reset 0 Flag 0: No reset occurred 1: Reset has occurred |
| 6 | Reserved | When read, an undefined value is returned. |
| 5 | SRESF4 | ECM System Reset Flag 0: No reset occurred 1: Reset has occurred |
| 4 | Reserved | When read, the value after reset is read. |
| 3 | SRESF2 | Software System Reset 0 Flag 0: No reset occurred 1: Reset has occurred |
| 2 | SRESF1 | CVM Reset Flag This flag is only cleared by Power On Reset, Debugger Initiated Reset, or software. 0: No reset occurred 1: Reset has occurred |

Table 8.6 Reset Factor Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 1 | SRESF0 | Pin Reset Flag This flag is also set by Power On Reset and Debugger Initiated Reset. This flag is only cleared by software. 0: No reset occurred 1: Reset has occurred |
| 0 | PRESF0 | Power On Reset Flag This flag is also set by Debugger Initiated Reset. This flag is only cleared by CVM Reset or software. 0: No reset occurred 1: Reset has occurred |

8.3.2 RESFC — Reset Factor Clear Register

This register clears the reset flags of the RESF register.

This register is always read as 00_H.

Access: This register can be written in 32-bit units.

Address: FFF8 1008_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|---------|---------|----|---------|----|---------|----|---------|---------|---------|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | ARESFC3 | ARESFC2 | — | ARESFC0 | — | SRESFC4 | — | SRESFC2 | SRESFC1 | SRESFC0 | PRESFC0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | W | W | R | W | R | W | R | W | W | W | W |

Table 8.7 Reset Factor Clear Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 11 | Reserved | When writing, write the value after reset. |
| 10 | ARESFC3 | Filed BIST Execution Flag Clear 0: Do not clear flag 1: Clear flag |
| 9 | ARESFC2 | ECM Application Reset Flag Clear 0: Do not clear reset flag 1: Clear reset flag |
| 8 | Reserved | When writing, write the value after reset. |
| 7 | ARESFC0 | Software Application Reset 0 Flag Clear 0: Do not clear reset flag 1: Clear reset flag |
| 6 | Reserved | When writing, write the value after reset. |
| 5 | SRESFC4 | ECM System Reset Flag Clear 0: Do not clear reset flag 1: Clear reset flag |
| 4 | Reserved | When writing, write the value after reset. |
| 3 | SRESFC2 | Software System Reset 0 Flag Clear 0: Do not clear reset flag 1: Clear reset flag |
| 2 | SRESFC1 | CVM Reset Flag Clear 0: Do not clear reset flag 1: Clear reset flag |
| 1 | SRESFC0 | Pin Reset Flag Clear 0: Do not clear reset flag 1: Clear reset flag |
| 0 | PRESFC0 | Power On Reset Flag Clear 0: Do not clear reset flag 1: Clear reset flag |

8.3.3 SWSRESA0 — Software System Reset Request Register 0

This register is used to generate a System Reset 2.

This register is always read as 00_H.

Access: This register can be written in 32-bit units.

Address: FFF8 1100_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SWSRESA0_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 8.8 Software System Reset Request Register 0 Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SWSRESA0_0 | Software System Reset Trigger 0 This bit is read as 0. 0: no function 1: generate Software System Reset 2 |

8.3.4 SWARESA0 — Software Application Reset Request Register 0

This register is used to generate an Application Reset 1.

This register is always read as 00_H.

Access: This register can be written in 32-bit units.

Address: FFF8 1200_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SWARE SA0_0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 8.9 Software Application Reset Request Register 0 Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SWARESA0_0 | Software Application Reset Trigger 0 This bit is read as 0. 0: no function 1: generate Software Application Reset 1 |

8.3.5 RESC — Reset Configuration Register

This register selects reset categories that ECM Reset generates.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2800_H

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RESC0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 8.10 Reset Configuration Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | RESC0 | ECM Reset Configuration 0: ECM Reset Generates a System Reset 2 1: ECM Reset Generates an Application Reset 1 |

8.3.6 STAC_DTSRAM — RAM Initialization Mode Control Register for DTS RAM

This register is used to control the RAM initialization execution of the DTS RAM. In Application Reset 1, initialization of DTS RAM is executed depending on this register.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1320_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RZERO MD1 | RZERO MD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 8.11 RAM Initialization Mode Control Register for DTS RAM Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | RZEROMD[1:0] | RAM Initialization Mode for DTS RAM x0: Disabled 01: Prohibited 11: Enabled |

8.3.7 STAC_GRAM — RAM Initialization Mode Control Register for Global RAM

This register is used to control the RAM initialization execution of the Global RAM. In Application Reset 1, initialization of Global RAM is executed depending on this register.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1420_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|-----------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RZERO MD1 | RZERO MD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 8.12 RAM Initialization Mode Control Register for GRAM Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | RZEROMD[1:0] | RAM Initialization Mode for Global RAM x0: Disabled 01: Prohibited 11: Enabled |

8.3.8 STAC_LM0 — RAM Initialization Mode Control Register for Local RAM

This register is used to control the RAM initialization execution of the Local RAM. In System Reset 1 (except CVM reset), System Reset 2, and Application Reset 1, initialization of Local RAM is executed depending on this register.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1520_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RZERO MD1 | RZERO MD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 8.13 RAM Initialization Mode Control Register for PE1 Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | RZEROMD[1:0] | RAM Initialization Mode for Local RAM x0: Disabled 01: Prohibited 11: Enabled |

8.3.9 STAC_LM10 — RAM Initialization Mode Control Register for CSIH

This register is used to control the RAM Initialization execution of the CSIH. In Application Reset 1, initialization of RAM for CSIH is executed depending on this register.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1E20_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|--------------|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RZERO MD1 | RZERO MD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 8.14 RAM Initialization Mode Control Register for CSIH Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | RZEROMD[1:0] | RAM Initialization Mode for CSIH x0: Disabled 01: Prohibited 11: Enabled |

8.4 Operation

8.4.1 Reset Categories

Power On Reset

The Power On Reset represents a Start-Up of the device during power up. This reset occurs when VCC supply voltage is under a specified level and is released when VCC supply voltage is over the specified level. In this case entire microcontroller is initialized. Power On Reset can also be triggered by Debugger.

System Reset 1

System Reset 1 is identical to the Power On Reset with the following exceptions. The Reset Factor Register of the Reset Controller and the Error Source Status Register of the Error Control Module and a part of Core Voltage Monitor registers are excluded from this reset. System Reset 1 sources are as follows.

- Pin Reset
- Core Voltage Monitor Reset
- Debugger Disconnect Reset

System Reset 2

System Reset 2 is identical to the System Reset 1 with the following exceptions. Reset Configuration register for ECM is excluded from reset. System Reset 2 sources are as follows.

- Software Reset by writing the SWSRESA0 register.
- Error Control Module Reset (The Reset Category is configured by the Reset Configuration Register)

Application Reset 1

Application Reset 1 is used for fast re-initialization of the application. This reset is basically identical to the System Reset 2 with the following exceptions. The Option bytes stored in the FLASH is not reloaded. Filed BIST is not executed too. Application Reset 1 sources are as follows.

- Software Reset by writing the SWARES0 register.
- Error Control Module Reset (The Reset Category is configured by the Reset Configuration Register)

8.4.2 Reset Sources

Power On Reset

The Power On Reset refers to a Start-Up of the device during power up. The power up condition is detected by a Power-On-Clear circuit (POC). It continuously compares the power supply voltage VCC with an internal reference voltage (V_{poc}). If VCC lowers below the internal reference voltage ($VCC < V_{poc}$), Power On Reset is generated. For details, refer to **Section 37, Electrical Characteristics**.

Pin Reset (by $\overline{\text{RESET}}$)

A dedicated Reset Input pin ($\overline{\text{RESET}}$) is available for initialization.

The associated buffer has input hysteresis. Therefore, there is no restriction for the slew rate of the reset signal as defined in the Electrical Characteristics.

The Pin Reset is active low. Its status will propagate independently from clock activity.

In case of an open reset input, the device will be initialized via the internal pull-down of the Reset Input Buffer. This ensures that the device is always in a safe state.

A Reset glitch filter is provided. For details about the filter characteristics, please refer to **Section 37, Electrical Specifications**.

In case of Power-up and Power-down, $\overline{\text{RESET}}$ should be low level. In case of Power-up, $\overline{\text{RESET}}$ must be asserted until VDD exceeds the threshold and main oscillator is stabilized. Until VCC and VDD exceed the threshold, Internal reset is maintained. After Internal reset and $\overline{\text{RESET}}$ are released, Filed BIST and RAM initialization are executed. For details on the required $\overline{\text{RESET}}$ low level period ($t_{DVDDPUR}$, $t_{DVCCPUR}$) and the threshold of VCC and VDD please refer to **Section 37, Electrical Specifications**.

In case of Pin Reset excluding power-up/down, the low level pulse width of the $\overline{\text{RESET}}$ input must be greater than the value of noise-filter characteristic (t_{WRSL}) to activate the reset. For details on the required $\overline{\text{RESET}}$ low level pulse width, please refer to **Section 37, Electrical Specifications**.

Core Voltage Monitor Reset

The Core Voltage Monitor reset can be asserted as soon as one of the core supply voltages is outside the operating range. The execution of the CVM reset can be masked as well. For details, refer to **Section 10, Core Voltage Monitor**.

The CVM operation is independent from the Pin Reset.

ECM Reset

No device failure (e.g. illegal access) will result in a reset. All failures which are detectable by the Error Control Module (ECM) generate dedicated internal reset or interrupts (FENMI or EINT) instead. The failure signals and system behavior can be configured in the ECM (see **Section 32, Error Control Module (ECM)**). Optionally the device failures can be configured to be used as a Reset Source.

The ECM Reset can generate a System Reset 2 or an Application Reset 1. The behavior can be configured by the Reset Configuration Register.

Software Reset

The device supports multiple software resets, and resets from different reset categories can be triggered.

- Software System Reset Request Register 0 (SWSRESA0) triggers a System Reset 2
- Software Application Reset Request Register 0 (SWARESAS0) triggers an Application Reset 1

Debug Initiated Reset

Transits to the internal reset state when the microcontroller is connected with the debug tool and receives the reset command from the debugger. It can trigger a Power On Reset. For details, refer to the **Section 34, On-Chip Debugging Unit (OCD)**.

Debugger Disconnect Reset

System reset 1 is generated when the input level of debug reset pin ($\overline{\text{DCUTRST}}$ pin) is changed from high to low. (e.g. disconnecting of debug tool).

8.4.3 Reset Flags

Any reset source can be identified by software in the Reset Factor Register (RESF). Each reset source will be indicated by one bit when it occurs. The bit status can be read out after the reset has been executed.

The Power On Reset flag is set by Power On Reset or Debugger Initiated Reset, and can only be cleared by a CVM Reset or by software.

The reset factor which sets or clears all reset flags except the Power On Reset flag is different for each flag. For details, see **8.3.1, RESF — Reset Factor Register**.

8.4.4 Reading Option Bytes from FLASH

Reading Option Bytes from FLASH memory is executed by Power On Reset, System Reset 1 and System Reset 2.

8.4.5 Filed BIST

Filed BIST is executed by Power On Reset, System Reset 1 and System Reset 2. In System Reset 1 (Pin reset only) and System Reset 2, Filed BIST execution can be disabled by the Field BIST Control Register (BSEQ0CTL). For details of BSEQ0CTL register, see **Section 31.6, BIST**.

8.4.6 RAM initialization

RAM initialization to 0 is basically executed by all reset categories. For System Reset 1 (Pin reset only), System Reset2 and Application Reset 1, Initialization of Local RAM can be disabled by the RAM Initialization Mode Control Register.

For Application Reset 1, Initialization of DTS RAM, Global RAM, and CSIH RAM can be disabled by the RAM Initialization Mode Control Registers corresponding to the RAMs.

8.4.7 Reset Mask function

In debug mode, System Resets 1 and 2 and Application Reset 1 can be masked by debugger setting. For details, please refer to **Section 34, On-Chip Debugging Unit (OCD)**.

Masked reset sources are shown in **Table 8.15**.

Table 8.15 Masked Reset Sources

| Reset Categories | Reset Sources |
|---------------------|-----------------------------|
| System Reset 1 | Pin Reset |
| System Reset 2 | ECM Reset (RESC0 = 0) |
| | Software Reset by SWSRESA0 |
| Application Reset 1 | ECM Reset (RESC0 = 1) |
| | Software Reset by SWARESAS0 |

8.4.8 Reset Output ($\overline{\text{RESETOUT}}$)

Any of the reset resources is generated, a reset output ($\overline{\text{RESETOUT}}$) is externally output. A reset output can be used to reset the external device at the same timing of reset generation in the microcontroller.

The figure below shows $\overline{\text{RESETOUT}}$ output operation.

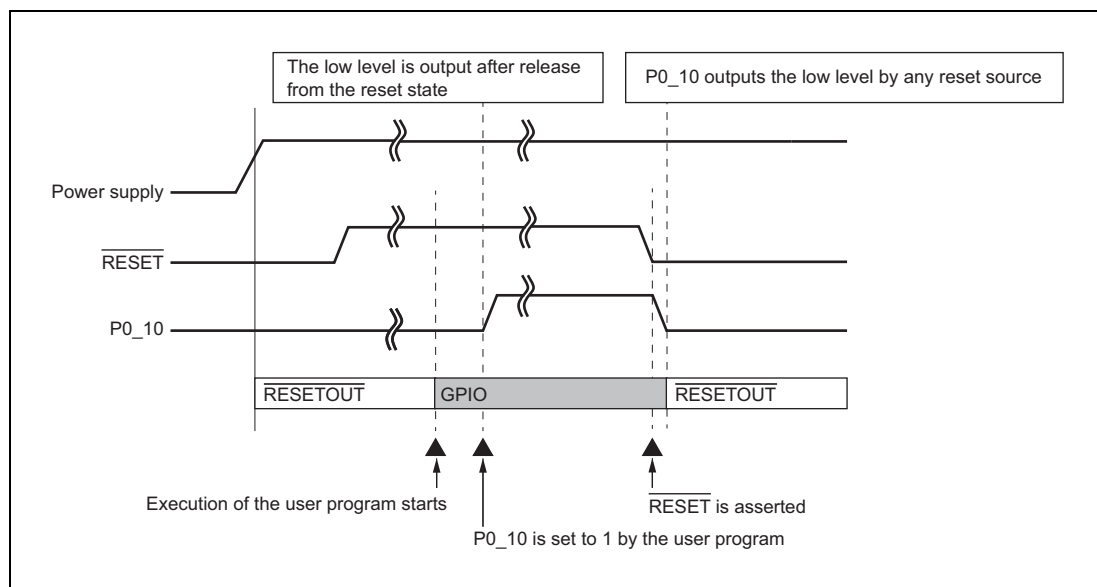


Figure 8.1 P0_10 Pin ($\overline{\text{RESETOUT}}$ Signal) Operation during Reset and after Release

Section 9 Power Supply Circuit

9.1 Features

This section describes the external voltage connection and internal voltage distribution required to operate the microcontroller. The power supply circuit has a POC (Power On Clear) circuit for safe startup.

The power supply construction of this device is classified into 2 categories.

Table 9.1 List of Power Type

| Item | Type Name | P1M-E |
|---|----------------------------------|-------|
| Single power supply (with internal regulators) | embedded Voltage Regulator (eVR) | √ |
| Dual Power Supply (DPS) | DPS | √ |

9.2 External Pin List

Table 9.2 shows the list of external pins

Table 9.2 List of External Pins

| Pin Name | I/O | Range | Function | eVR | DPS |
|----------|--------|----------------|--|-----|-----|
| E0VCC | Power | 3.0 to 5.5V | I/O power supply | √ | √ |
| E1VCC | | | | | |
| E0VSS | Ground | 0V | | √ | √ |
| E1VSS | | | | | |
| VCC | Power | 3.0 to 5.5V | POC, CVM, PLL, Oscillator, Flash, Internal voltage regulator power | √ | √ |
| VCL | Output | — | Capacitor connection pin for internal voltage regulator | √ | — |
| VDD | Power | 1.20V to 1.35V | Power supply for internal | — | √ |
| VSS | Ground | 0V | | √ | √ |
| A0VCC | Power | 3.0V to 5.5V | Power supply for A/D | √ | √ |
| A1VCC | | | | | |
| A0VSS | Ground | 0V | | √ | √ |
| A1VSS | | | | | |
| A0VREFH | Input | 3.0V to 5.5V | Reference voltage for A/D | √ | √ |
| A1VREFH | | | | | |

9.3 Block Diagram

9.3.1 Single Power Supply

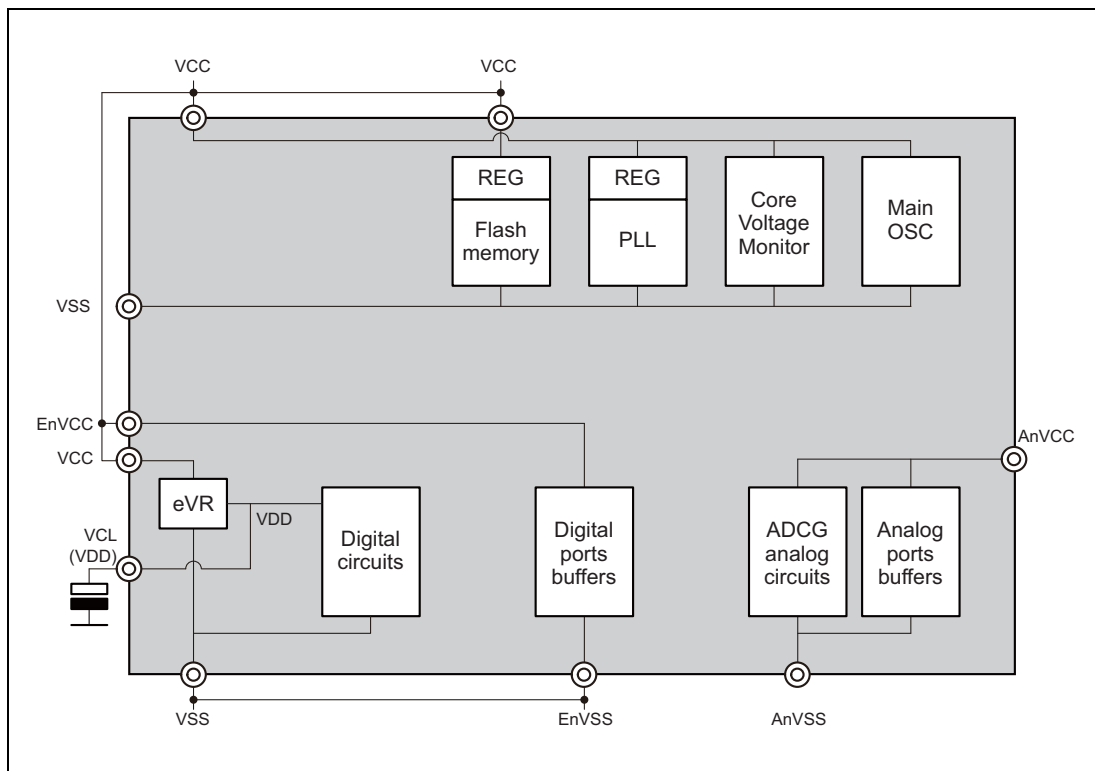


Figure 9.1 Single Power Supply Block Diagram

9.3.2 Dual Power Supply

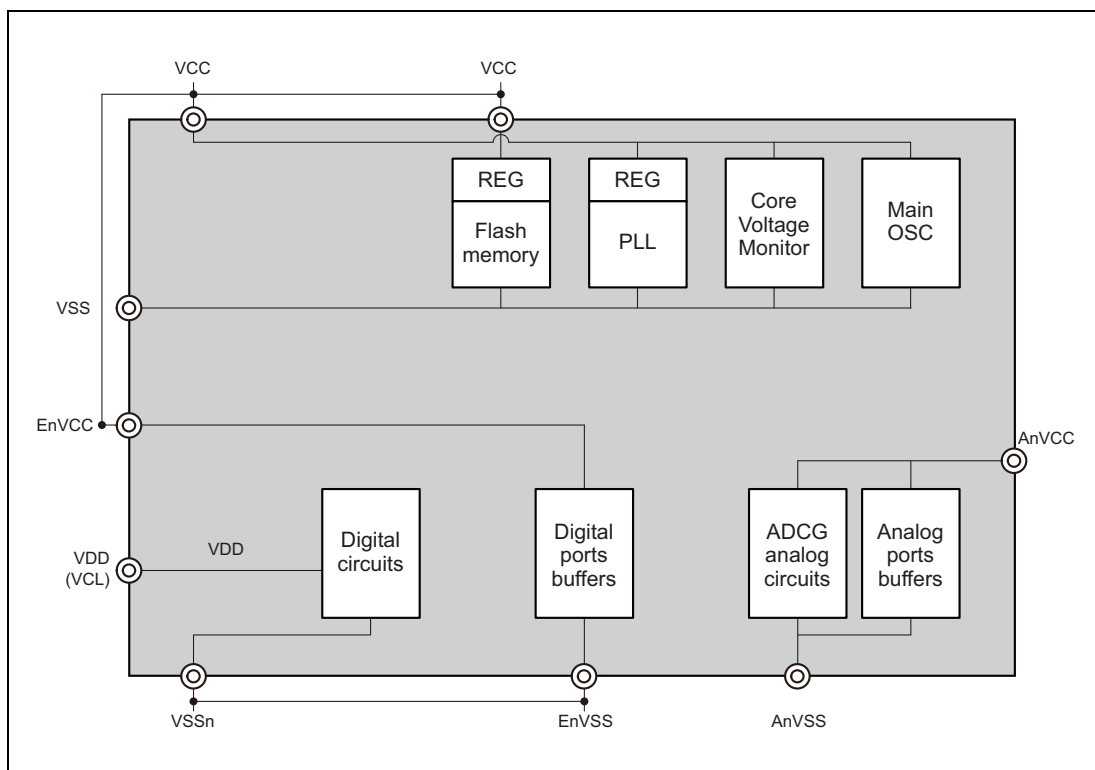


Figure 9.2 Dual Power Supply Block Diagram

9.4 Connection Example

9.4.1 Single Power Supply

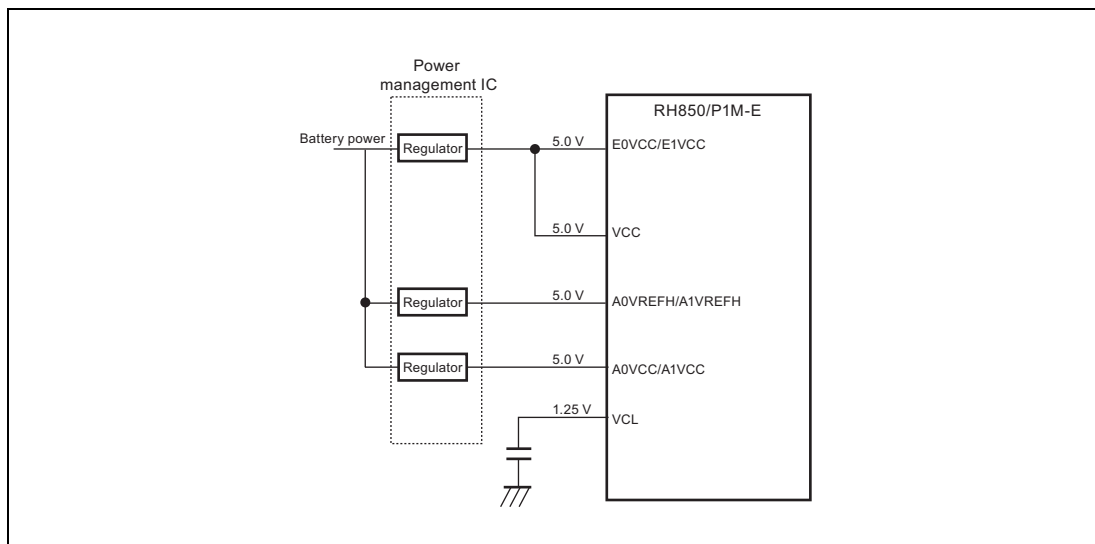


Figure 9.3 Power Management IC Connection Example (in Single-Power Supply Configuration)

9.4.2 Dual Power Supply

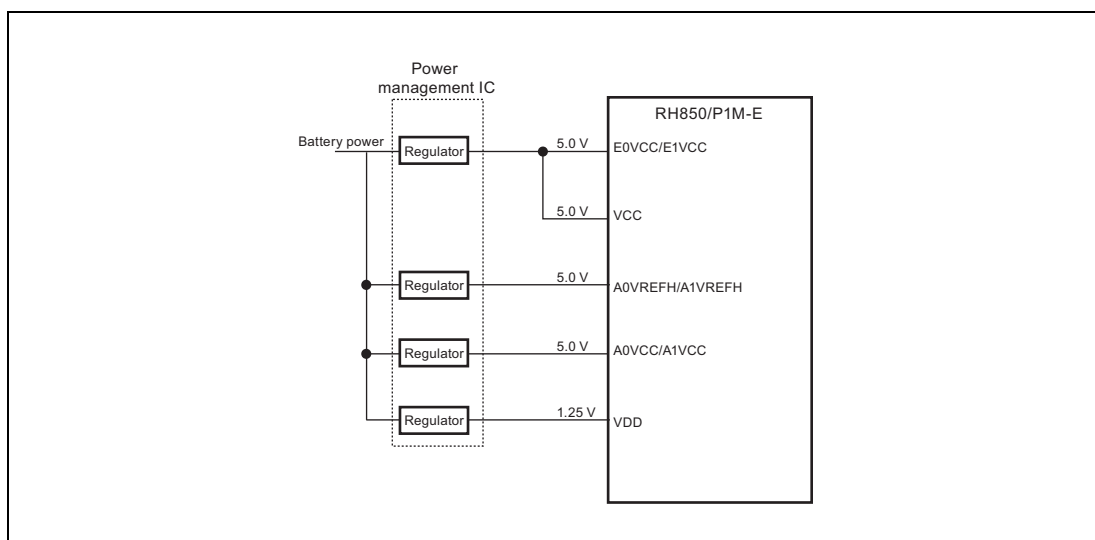


Figure 9.4 Power Management IC Connection Example (in Dual-Power Supply Configuration)

9.5 Power Up/Down Timing

See **Section 37, Electrical Specifications.**

Section 10 Core Voltage Monitor

10.1 Overview

10.1.1 Functional Overview

Core voltage monitor (CVM) monitors the VDD electric potential supplied to the core logic.

Deviation from operating range of the core voltage is informed by:

- Outputs a low level from the $\overline{\text{CVMOUT}}$ pin.
- Sets flags independently in the CVMF register at events (high voltage detection and low voltage detection).
- Able to reset of core voltage operation area by CVM Reset.

The CVM function is independent of the resets except Power On Reset.

All writable CVM control registers are protected by P-Bus guard.

Diagnostic function:

- High voltage error and low voltage error can be generated without influencing core voltage itself by changing reference voltage.
- The signal path to the $\overline{\text{CVMOUT}}$ pin can optionally be masked at diagnostic.
- $\overline{\text{CVMOUT}}$ pin provides read back function to check pin level.

10.1.2 Block Diagram

The block diagram of CVM for DPS is shown in **Figure 10.1**.

The block diagram of CVM for eVR is shown in **Figure 10.2**.

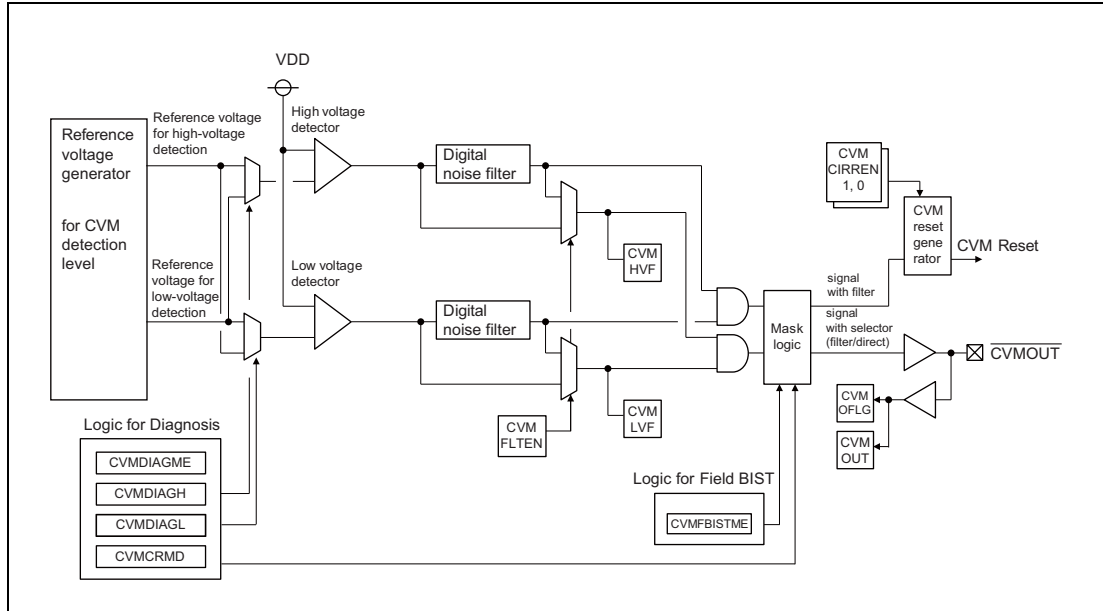


Figure 10.1 Block diagram of CVM for DPS

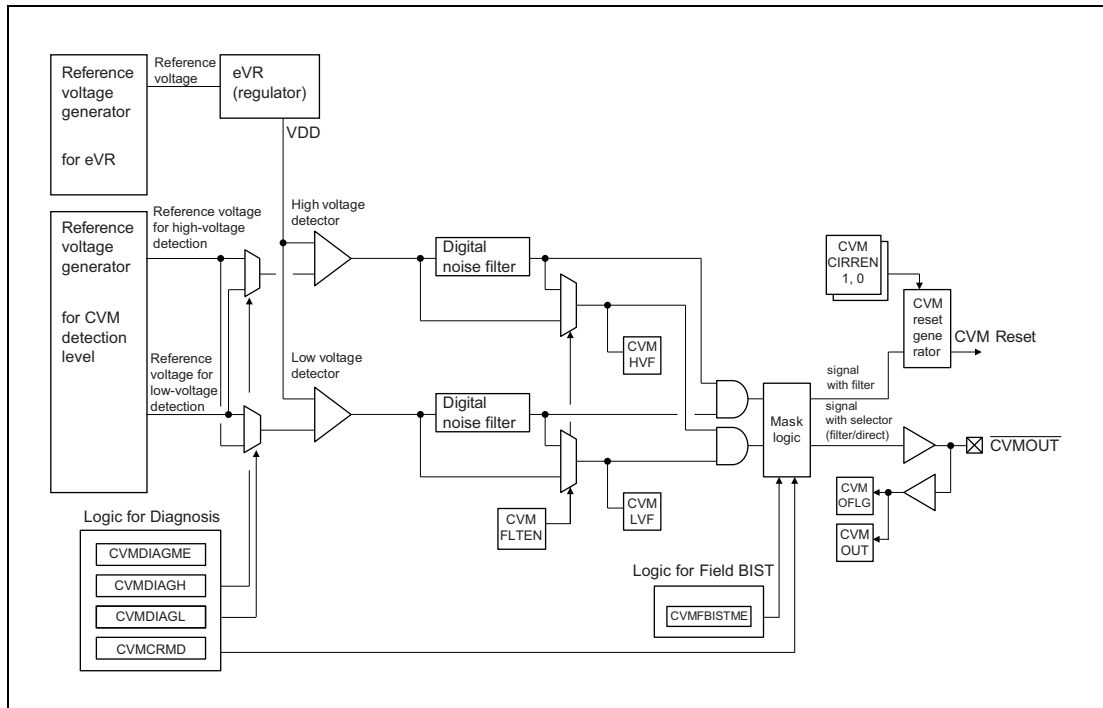


Figure 10.2 Block diagram of CVM for eVR

10.2 Input/Output Pins

I/O pin related to CVM is shown in **Table 10.1**.

Table 10.1 I/O Pins

| Pin Function Name | Direction | Description |
|-------------------|-----------|--|
| CVMOUT | Output | CVM internal voltage error detection output signal |

10.3 Registers

10.3.1 List of Registers

A register list related to CVM is shown in **Table 10.2**.

Table 10.2 List of Registers

| Address | Register Name | Description | Access Width | Value after Reset |
|------------------------|---------------|---|--------------|-------------------------------|
| FFF8 2C00 _H | CVMF | CVM Factor register | 8 | 00 _H |
| FFF8 2C04 _H | CVMDDE | CVM Detection Enable register | 8 | 03 _H |
| FFF8 2C0C _H | CVMDMASK | CVM Detection Output Mask Register | 8 | 00 _H |
| FFF8 2C10 _H | CVMDIAG | CVM DIAG mode setting register | 8 | 00 _H |
| FFF8 2C14 _H | CVMMON | CVM Monitor register | 8 | 0x _H ^{*2} |
| FFF8 2C18 _H | CVMFC | CVM Factor Clear Register | 8 | 00 _H |
| FFF8 2C1C _H | CVMDEW | CVM Detection Enable Set Register ^{*1} | 8 | 00 _H |

Note 1. Writing is permitted only once after Power On Reset.

Note 2. This initial value depends on the status of $\overline{\text{CVMOUT}}$ pin.

Register reset conditions are shown in **Table 10.3**.

Table 10.3 Register Reset Conditions

| Register Name | Reset condition | | | |
|---------------|-----------------|-----------------|----------------|---------------------|
| | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| CVMF | √ | | | |
| CVMDDE | √ | | | |
| CVMDMASK | √ | √ ^{*1} | | |
| CVMDIAG | √ | √ | | |
| CVMMON | | | | |
| CVMFC | | | | |
| CVMDEW | | | | |

Note 1. Pin Reset only.

10.3.2 CVMF — CVM Factor Register

The CVMF register indicates the state of $\overline{\text{CVMOUT}}$ pin output and the state of the detection of errors by the CVM.

This register is initialized only by Power On Reset.

Each flag of CVMF can be also cleared individually by writing “1” to the corresponding bit in CVMFC.

Access: This register can be read in 8-bit units.

Address: FFF8 2C00_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------|---|---|---|---|---|--------|--------|
| | CVMOFLG | — | — | — | — | — | CVMHVF | CVMLVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 10.4 CVMF Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 | CVMOFLG | <u>CVMOUT Flag</u> When $\overline{\text{CVMOUT}}$ pin has changed to low level, this flag is set to 1. 0: $\overline{\text{CVMOUT}}$ pin has not changed to low level. 1: $\overline{\text{CVMOUT}}$ pin has changed to low level. |
| 6 to 2 | Reserved | When read, the value after reset is read. |
| 1 | CVMHVF | High Voltage Detection Flag 0: No high core voltage violation detected. 1: High core voltage violation occurred. |
| 0 | CVMLVF | Low Voltage Detection Flag 0: No low core voltage violation detected. 1: Low core voltage violation occurred. |

10.3.3 CVMFC — CVM Factor Clear Register

CVMFC is a register to clear the corresponding bits in CVMF register.

The read value of this register is always 00_H.

Access: This register can be written in 8-bit units.

Address: FFF8 2C18_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|---|---|---|---|---|---------|---------|
| | CVMOFLGC | — | — | — | — | — | CVMHVFC | CVMLVFC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | R | R | R | R | R | W | W |

Table 10.5 CVMFC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | CVMOFLGC | Clears $\overline{\text{CVMOU}}\overline{\text{T}}$ Flag 0: Writing 0 has no effect. 1: Writing 1 clears CVMOFLG. |
| 6 to 2 | Reserved | When writing, write the value after reset. |
| 1 | CVMHVFC | Clears High Voltage Detection Flag 0: Writing 0 has no effect. 1: Writing 1 clears CVMHVF. |
| 0 | CVMLVFC | Clears Low Voltage Detection Flag 0: Writing 0 has no effect. 1: Writing 1 clears CVMLVF. |

10.3.4 CVMDE — CVM Detection Enable Register

CVMDE is a read-only register to control CVM reset, masking of the CVM signal, and filtering of the CVM signal.

This register is initialized by Power On Reset and reflect the settings written to the corresponding bits of CVMDEW.

Access: This register can be read in 8-bit units.

Address: FFF8 2C04_H

Value after reset: 03_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|---|------------|------------|----------|-----------|---|---|
| | CVMCIRREN1 | — | CVMCIRREN0 | CVMFBISTME | CVMFLTEN | CVMDIAGME | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R |

Table 10.6 CVMDE Register Contents

| Bit Position | Bit Name | Function |
|--|------------|---|
| 7 | CVMCIRREN1 | Enables CVM Reset. CVMCIRREN1, 0 1 _B : CVM internal reset is enabled. Other than the above: CVM internal reset is disabled. |
| 6 | Reserved | When read, the value after reset is read. |
| 5 | CVMCIRREN0 | Refer to bit 7 (CVMCIRREN1). |
| 4 | CVMFBISTME | $\overline{\text{CVMOUT}}$ output and CVM Reset Mask Control during Field BIST Operation 0: Masking of $\overline{\text{CVMOUT}}$ output and CVM reset during Field BIST Operation is enabled. 1: Masking of $\overline{\text{CVMOUT}}$ output and CVM reset during Field BIST Operation is disabled. |
| CAUTION | | |
| This bit setting is valid only for the Field BIST that started up from System Reset 1. $\overline{\text{CVMOUT}}$ output and CVM Reset are not masked if the Field BIST is started by System Reset 2, regardless of the setting of this bit. | | |
| 3 | CVMFLTEN | Enables Output Filter for $\overline{\text{CVMOUT}}$ 0: Output filter for $\overline{\text{CVMOUT}}$ is enabled. 1: Output filter for $\overline{\text{CVMOUT}}$ is disabled. |
| 2 | CVMDIAGME | Enables DIAG Function 0: DIAG function of the CVM is enabled. 1: DIAG function of the CVM is disabled. |
| 1, 0 | Reserved | When read, the value after reset is read. |

10.3.5 CVMDEW — CVM Detection Enable Set Register

CVMDEW is a register to set values of CVMDE.

Writing is permitted only once after Power On Reset is released. Subsequent write operation is ignored. The read value of this register is always 00_H.

Access: This register can be written in 8-bit units.

Address: FFF8 2C1C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------|---|--------------|--------------|-----------|------------|---|---|
| | CVM CIRREN1W | — | CVM CIRREN0W | CVM FBISTMEW | CVMFLTENW | CVMDIAGMEW | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | R | W | W | W | W | R | R |

Table 10.7 CVMDEW Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 7 | CVMCIRREN1W | The value written in this bit is set to CVMDE.CVMCIRREN1. |
| 6 | Reserved | When writing, write the value after reset. |
| 5 | CVMCIRREN0W | The value written in this bit is set to CVMDE.CVMCIRREN0. |
| 4 | CVMFBISTMEW | The value written in this bit is set to CVMDE.CVMFBISTME. |
| 3 | CVMFLTENW | The value written in this bit is set to CVMDE.CVMFLTEN. |
| 2 | CVMDIAGMEW | The value written in this bit is set to CVMDE.CVMDIAGME. |
| 1, 0 | Reserved | When writing, write the value after reset. |

10.3.6 CVMDMASK — CVM Detection Output Mask Register

CVMDMASK is a register to mask $\overline{\text{CVMOUT}}$ output when CVMDE.CVMDIAGME = 0.

The setting of CVMDMASK is ignored when CVMDE.CVMDIAGME = 1. Writing is possible regardless of the CVMDIAGME setting.

This register is initialized by Power On Reset or System Reset 1 (Pin Reset only).

Access: This register can be read/written in 8-bit units.

Address: FFF8 2C0C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | CVMCRMD |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 10.8 CVMDMASK Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | CVMCRMD | This bit masks $\overline{\text{CVMOUT}}$ when CVMDE.CVMDIAGME = 0. 0: $\overline{\text{CVMOUT}}$ output are not masked. 1: $\overline{\text{CVMOUT}}$ output are masked. $\overline{\text{CVMOUT}}$ becomes high level and the CVM Reset don't occur. |

NOTE

The value of CVMDE.CVMCIRREN1, 0 is "00_B" while the CVM diagnosis application is carried out after Power On Reset. Therefore CVM reset has been already masked by CVMCIRREN1, 0 even if CVMCRMD isn't set to "1".

10.3.7 CVMDIAG — CVM DIAG Mode Setting Register

CVMDIAG forces CVM comparators to output errors.

This register is valid only when CVMDE.CVMDIAGME = 0. The setting of CVMDIAG is ignored when CVMDE.CVMDIAGME = 1. Writing is possible regardless of the CVMDIAGME setting.

This register is initialized by Power On Reset and System Reset 1.

Access: This register can be read/written in 8-bit units.

Address: FFF8 2C10_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|----------|----------|---|---|
| | — | — | — | — | CVMDIAGH | CVMDIAGL | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R | R |

Table 10.9 CVMDIAG Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | CVMDIAGH | This bit controls the detection level of high voltage detector. 0: CVM monitors at a reference for high-voltage detection. 1: High voltage violation is forcibly generated by changing the detection level to reference voltage for low-voltage detection. |
| 2 | CVMDIAGL | This bit controls the detection level of low voltage detector. 0: CVM monitors at a reference for low-voltage detection. 1: Low voltage violation is forcibly generated by changing the detection level to reference voltage for high-voltage detection. |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

10.3.8 CVMMON — CVM Monitor Register

CVMMON register reflects the status of $\overline{\text{CVMOUT}}$ pin.

Access: This register can be read in 8-bit units.

Address: FFF8 2C14_H

Value after reset: 0X_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|--------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | CVMOUT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | — |
| R/W | R | R | R | R | R | R | R | R |

Table 10.10 CVMMON Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | CVMOUT | $\overline{\text{CVMOUT}}$ Pin Level 0: $\overline{\text{CVMOUT}}$ Output level is at low level. 1: $\overline{\text{CVMOUT}}$ Output level is at High level. |

10.4 Operation

10.4.1 CVM Basic Function

CVM constantly monitors the VDD voltage except when VCC is not stable and Power On Reset occurs.

CVM has high and low voltage detection circuit. If it detects voltage error, it records the error to CVMF register and notifies the error to the outside via $\overline{\text{CVMOUT}}$ pin. It also generates internal reset (CVM reset) triggered by detection of voltage error.

(1) CVM detection flag

CVM sets “1” to CVMF.CVMHVF when VDD becomes higher than high voltage detection level of CVM. Similarly, CVM sets “1” to CVMF.CVMLVF when VDD becomes lower than low voltage detection level of CVM.

These flags can be cleared to “0” by writing in the corresponding bits of CVMFC (CVMHVFC for CVMHVF, CVMLVFC for CVMLVF). These flags are also cleared by Power On Reset (not cleared by other resets).

Figure 10.3 shows the operation example of CVMHVF and CVMLVF.

(2) $\overline{\text{CVMOUT}}$ pin

While VDD voltage is higher than high voltage detection level or lower than low voltage detection level of CVM, $\overline{\text{CVMOUT}}$ pin outputs a low level signal. If CVM does not detect a voltage error, the $\overline{\text{CVMOUT}}$ pin outputs a high level signal.

To avoid an unexpected operation of CVM circuit caused by instable power supply at power up, $\overline{\text{CVMOUT}}$ is fixed to low during Power On Reset period. $\overline{\text{CVMOUT}}$ outputs error detection result at all other times.

The status of $\overline{\text{CVMOUT}}$ can be read from CVMF.CVMOFLG or CVMMON.CVMOUT. CVMOFLG is set to “1” when $\overline{\text{CVMOUT}}$ pin becomes low. It is cleared to “0” when “1” is written to CVMFC.CVMOFLGC. It is also cleared by Power On Reset (not cleared by other resets).

Figure 10.3 shows the operation example of $\overline{\text{CVMOUT}}$ and CVMOFLG.

CVMMON.CVMOUT is used for self-diagnosis function. See **Section 10.4.3, CVM Diagnosis Function** for further information.

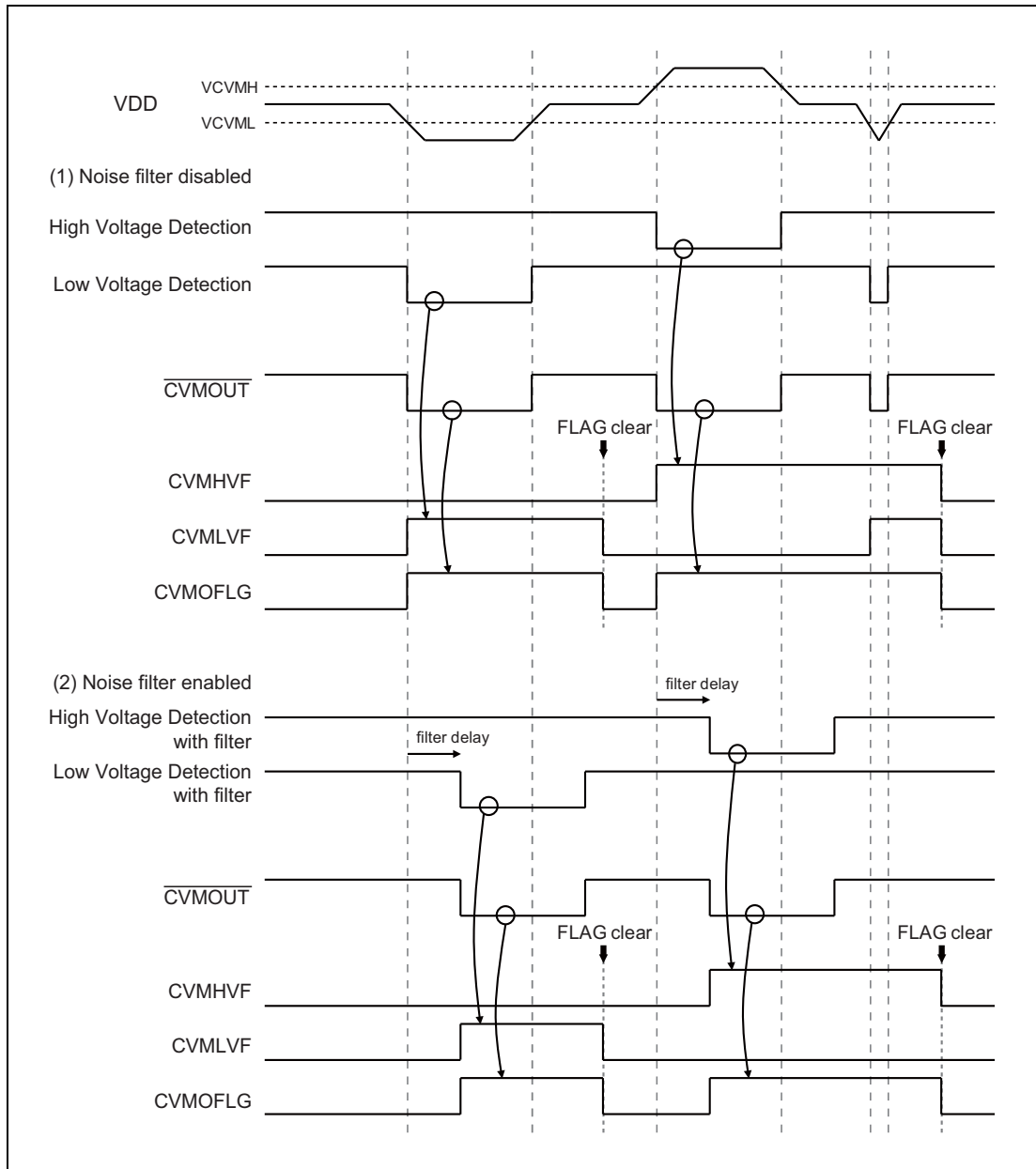


Figure 10.3 CVM Basic Operation

(3) CVM reset

If VDD voltage becomes higher than high voltage detection level or lower than low voltage detection level of CVM when CVM Reset is enable (CVMDE.CVMCIRREN1, 0 = "11_B") in advance, CVM reset occurs and the MCU is initialized. CVM reset is released since CVM detects the normal voltage.

Figure 10.4 shows the operation example of CVM reset.

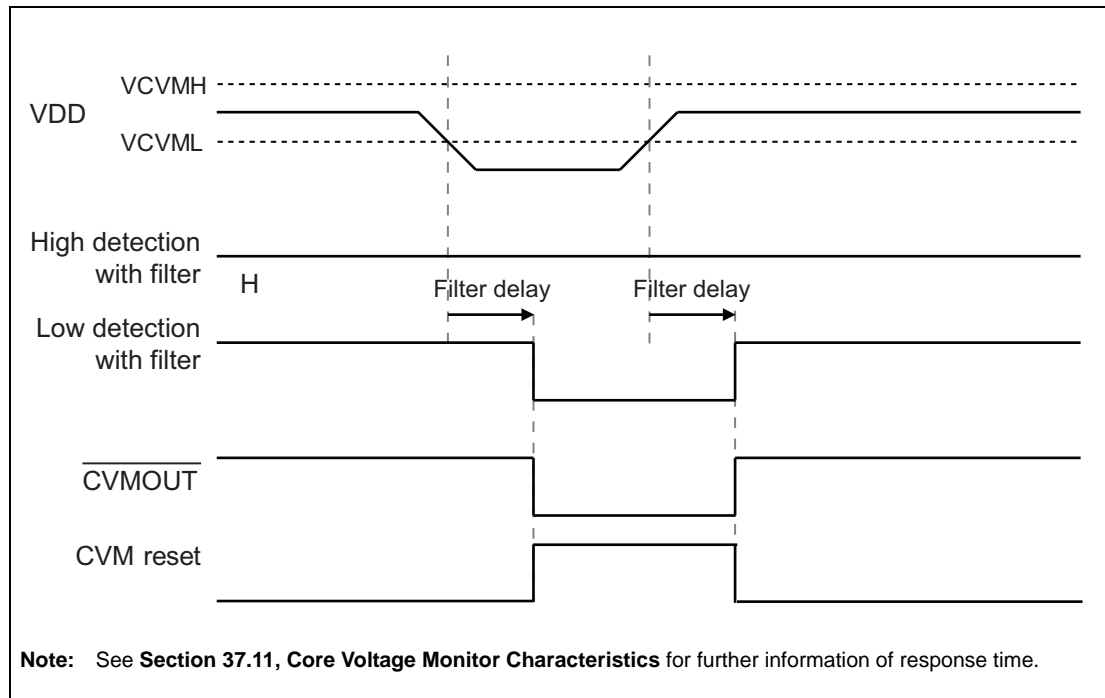


Figure 10.4 CVM Reset Operation

(4) Digital noise filter

The output signals from the CVM comparators pass through a digital noise filter to remove temporary glitches. Whether to use filtered output or to bypass the filter can be selected by CVMDE.CVMFLTEN. This setting is applied to $\overline{\text{CVMOUT}}$, CVMHVF and CVMLVF. For CVM reset, filtered signals are always used.

10.4.2 CVM Function in Field BIST operation and Serial Programming Mode (Mask of CVMOUT Pin and CVM Reset)

It is possible to mask $\overline{\text{CVMOUT}}$ output and CVM reset while Field BIST is being carried out after Power On Reset and System Reset 1 if $\text{CVMDE.CVMFBISTME} = 0$. In this case, $\overline{\text{CVMOUT}}$ is fixed to high level and a CVM reset does not occur. $\overline{\text{CVMOUT}}$ output is fixed to high level in serial programming mode as well.

10.4.3 CVM Diagnosis Function

(1) Change of CVM detection level for forcibly generating CVM error

When CVMDE.CVMDIAGME is "0", CVM detection level can be changed by CVMDIAG.CVMDIAGH or CVMDIAG.CVMDIAGL . CVM error detection signal can be intentionally generated by setting these registers even when VDD voltage is in the operating range.

(2) Mask of $\overline{\text{CVMOUT}}$ pin and CVM Reset

$\overline{\text{CVMOUT}}$ output can be masked by setting CVMDMASK.CVMCRMD to "1" if CVMDE.CVMDIAGME is "0". With this setting, $\overline{\text{CVMOUT}}$ is fixed to high level even when a CVM error is generated by CVMDIAGH or CVMDIAGL .

The value of CVMDE.CVMCIRREN1 , 0 is "00_B" while the CVM diagnosis application is carried out after Power On Reset. Therefore, CVM reset is already masked by CVMCIRREN1 , 0 even if CVMCRMD isn't set to "1".

(3) Monitor of $\overline{\text{CVMOUT}}$ pin

The status of $\overline{\text{CVMOUT}}$ pin can be monitored by CVMMON.CVMOUT . By reading this register, it can be checked whether $\overline{\text{CVMOUT}}$ is at low level when a CVM error is generated by CVMDIAGH or CVMDIAGL .

If VDD voltage is out of the operating range, the value read from the CVMMON register can't be guaranteed.

(4) The flow of CVM diagnosis application

CVMDE.CVMDIAGME is initialized to "0" by Power On Reset. Thus, the CVM diagnosis function is enabled at start-up after Power On Reset. The diagnostic function can be controlled by the user application by using CVMDMASK and CVMDIAG .

Figure 10.5 shows an example of the diagnosis application.

After diagnosis application is over, write "1" to $\text{CVMDEW.CVMDIAGMEW}^*1$ and disable CVM diagnosis function immediately. This function protects each flag against erroneous operation when VDD is out of the operating range.

Note 1. Write to the other bits of CVMDEW as well at the same time because writing to CVMDEW is permitted only once after Power On Reset.

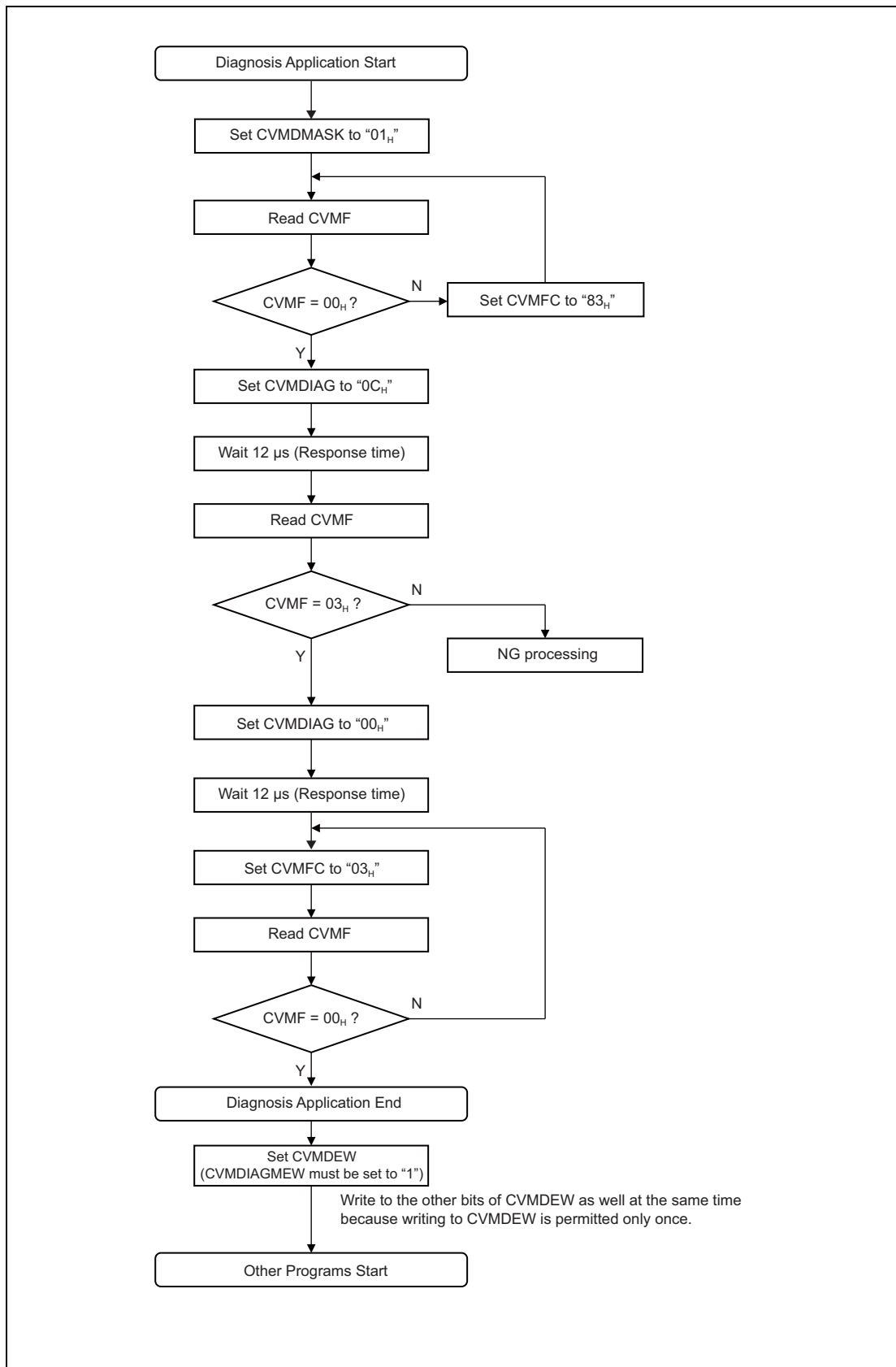


Figure 10.5 Example of Flowchart for CVM Diagnosis Application

10.5 Usage Notes

(1) Flag bits (CVMHVF, CVMLVF, CVMOFLG)

Setting 1 to CVMHVF by error detection has priority over clearing CVMHVF by CVMHVFC. Similarly, setting 1 to CVMLVF by error detection has priority over clearing CVMLVF by CVMLVFC. Furthermore, setting 1 to CVMOFLG by $\overline{\text{CVMOUT}}$ pin status has priority over clearing CVMOFLG by CVMOFLGC.

These functions protect each flag against erroneous operation when VDD is out of the operating range. When VDD is recovered to the operating range, the MCU needs to be initialized by Pin Reset or CVM Reset. Otherwise, the operation of the MCU may continue to be unstable and the flags may be unintentionally cleared.

(2) $\overline{\text{ERROROUT}}$ pin

Enable CVM Reset if you want to notify CVM error detection signal to outside of the MCU via the $\overline{\text{ERROROUT}}$ pin.

When CVM Reset occurs, $\overline{\text{ERROROUT}}$ pin becomes Hi-Z (high impedance) regardless of whether VDD is in the operating range or not.

Section 11 Temperature Sensor

The Temperature Sensor (TSN0) measures temperature by using one analog channel. TSN0 has one unit implemented.

11.1 Features of RH850/P1M-E Temperature Sensor

11.1.1 Number of Units

This product is provided with a temperature sensor with the following number of units.

Table 11.1 Number of Units

| Temperature Sensor | |
|--------------------|------|
| Number of units | 1 |
| Name | TSN0 |

11.1.2 Register Base Address

TSN0 register base addresses are listed in the following table.

The register addresses of the TSN0 are given as offsets from the base addresses.

Table 11.2 Register Base Address

| Base address Name | Base Address |
|-------------------|------------------------|
| TSN0_base | FFF9 3000 _H |

11.1.3 Clock Supply

Clock supply by and to the temperature sensor is listed in the following table.

Table 11.3 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|------------------------------------|
| TSN0 | PCLK | Low-speed peripheral clock CLK_LSB |

11.1.4 Interrupt Request

TSN0 has no interrupt request.

11.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

11.1.6 External Input/Output Signals

TSN0 has no external pin.

11.2 Overview

11.2.1 Functional Overview

TSN0 has the following functions and features.

- Temperature measurement
TSN0 measures temperature by using analog channel 12 (AN112) of A/D converter unit 1 (ADCG1).
Temperature measurement time (CLK_ADC = 40 MHz): 4 μ s (A/D conversion time of 1 μ s \times 4 times)
Temperature measurement time (CLK_ADC = 20 MHz): 45.2 μ s (A/D conversion time of 11.3 μ s \times 4 times)
Stabilization time: 200 μ s
- Temperature Error notification
A temperature error signal is sent to the ECM module when the temperature reaches an upper or lower limit.
- Temperature measurement modes
One-shot: A/D conversion is executed four times consecutively.
Continuous: Four consecutive virtual channels are configured for the temperature sensor and the corresponding scan group 4 is placed in multi-cycle scan mode.
- Self-diagnosis
Self-diagnosis through the insertion of high-temperature errors is supported.

11.2.2 Block Diagram

Figure 11.1 shows the block diagram of temperature sensor.

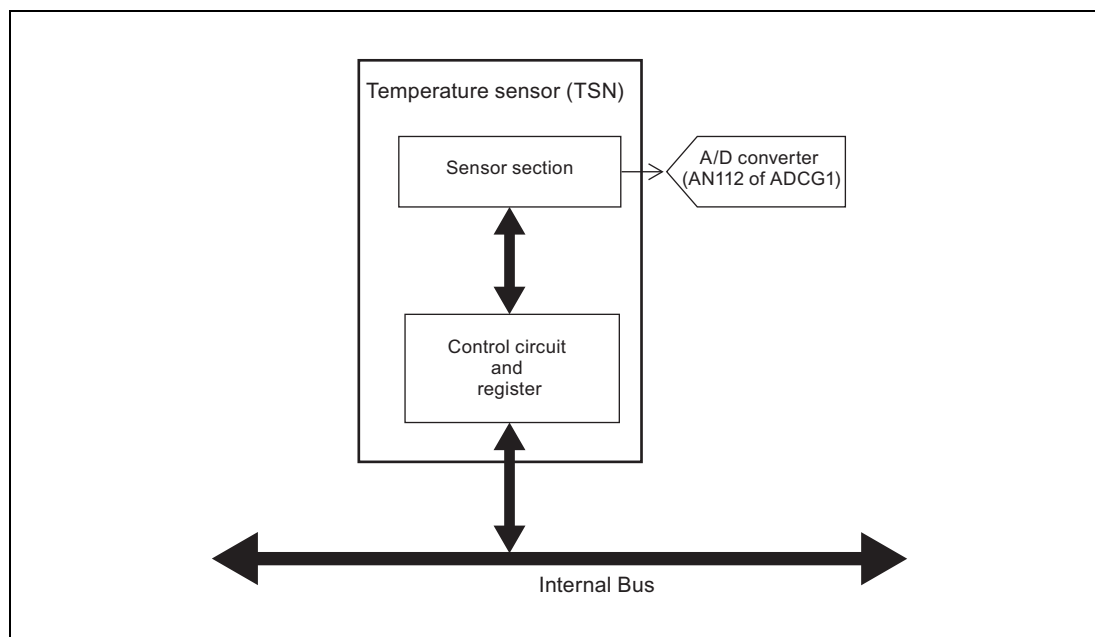


Figure 11.1 Block Diagram of Temperature Sensor

11.3 Register

11.3.1 Register List

The following table lists the registers.

For details about <TSN0_base>, see **Section 11.1.2, Register Base Address**.

Table 11.4 List of Registers

| Register Name | Symbol Name | R/W | Value after reset | Address |
|---|-------------|-----|------------------------|--------------------------------|
| Temperature sensor control register | TSN0CR | R/W | 0000 0000 _H | <TSNn_base> + 000 _H |
| Temperature sensor status register | TSN0STAT | R | 0000 0000 _H | <TSNn_base> + 004 _H |
| Temperature sensor diagnosis control register | TSN0DIAG | R/W | 0000 0000 _H | <TSNn_base> + 008 _H |
| Temperature sensor reference temperature storage register | TSNREFD | R | Undefined*1 | FFCD 019C _H |

Note 1. Fixed value is set at the time of shipping.

11.3.2 TSN0CR — Temperature Sensor Control Register

This register controls the temperature sensor. This register can be set up when the A/D converter is stopped.

Access: This register can be read/written in 32-bit units.

Address: <TSNn_base> + 000_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSNEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 11.5 TSN0CR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TSNEN | This bit controls the temperature sensor 0: The temperature sensor is disabled. 1: The temperature sensor is enabled. |

11.3.3 TSN0STAT — Temperature Sensor Status Register

This register indicates the status of the temperature sensor.

Access: This register can be read only in 32-bit units.

Address: <TSNn_base> + 004_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSNST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 11.6 TSN0STAT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | TSNST | This bit indicates the status of the temperature sensor 0: The temperature sensor is disabled. (Temperature sensor stabilization time 200 μs has not elapsed.) 1: The temperature sensor is enabled. (Temperature sensor stabilization time 200 μs has elapsed.) |

11.3.4 TSN0DIAG — Temperature Sensor Diagnosis Control Register

This register controls self-diagnosis of the temperature sensor. It can be configured when the A/D converter is stopped.

Access: This register can be read or written in 32-bit units.

Address: <TSNn_base> + 008_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSN SELF DIAG |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 11.7 TSN0DIAG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TSNSELFDIAG | This bit controls self-diagnosis of the temperature sensor. 0: The temperature sensor self-diagnosis is disabled. 1: The temperature sensor self-diagnosis is enabled. |

11.3.5 TSNREFD — Temperature Sensor Reference Temperature Storage Register

In this register, the measured value at the time of shipping is stored.

The reference temperature (at A1VREFH = 5.0 V, T_j = 150°C) is stored in TSNREFDH and the reference temperature (at A1VREFH = 5.0 V, T_j = 25°C) is stored in TSNREFDL.

Access: This register can be read only in 32-bit units.

Address: FFCD 019C_H

Value after reset: Undefined: (Fixed value is set at the time of shipping.)

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----------------|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | TSNREFDH[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | TSNREFDL[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 11.8 TSNREFD Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 31 to 28 | Reserved | When read, the value after reset is read. |
| 27 to 16 | TSNREFDH[11:0] | The value of the reference temperature measured at T _j = 150°C |
| 15 to 12 | Reserved | When read, the value after reset is read. |
| 11 to 0 | TSNREFDL[11:0] | The value of the reference temperature measured at T _j = 25°C |

11.4 Functions

11.4.1 Temperature Measurement

TSN0 measures temperatures by using the AN112 channel of ADCG1. The AN112 channel is set for scan group 4 of the ADCG1 and four consecutive virtual channels must be set for A/D conversion of the signal from the temperature sensor.

Refer to **Figure 11.2** for an example. The result of measurement is stored in the data register corresponding to the fourth virtual channel.

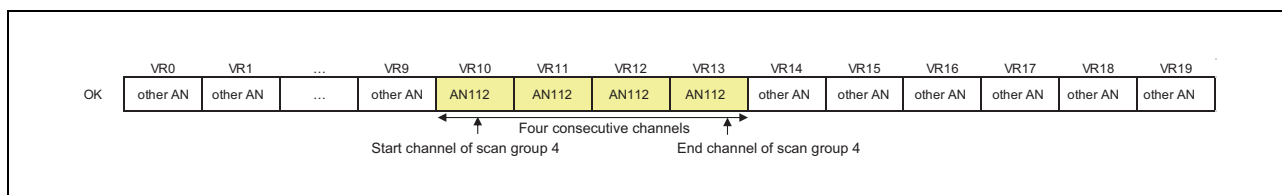


Figure 11.2 Example Setting for Temperature-Sensor-Dedicated Use of Conversion Channel AN112 of ADCG1

Set TSN0 up for measuring the temperature by making the register settings in the procedure below.

1. Set the TSN0-dedicated A/D conversion channel AN112 in four consecutive virtual channels of scan group 4.
2. Enable the temperature sensor by setting the TSNEN bit of the TSNCR register to “1”. This step can precede step 1 above.
3. Wait for TSN0 to start up. After setting the TSNEN bit of the TSNCR register to “1”, wait for 200 us before proceeding with the next step (step 4). This is the waiting time required for the temperature sensor to start up. Do not proceed with the next step until this time has elapsed. You can confirm that the sensor has started up by checking the TSNST bit of the TSNSTAT register.
4. TSN0 starts measuring temperature and a software or hardware trigger starts A/D conversion of scan group 4 of the ADCG1.
5. Check the result of measurement. The measurement of temperature is completed when A/D conversion of the fourth virtual channel is completed. The result of measuring the signal from the temperature sensor is stored in the data register corresponding to the fourth virtual channel. For conversion of the result to degrees Celsius, refer to **Section 11.5, Calculating the Temperature**.
6. When abnormal temperature notification by the temperature sensor is to be used, the upper and lower limit error bits of ADCG1 must be cleared. Since discharging proceeds in the first A/D conversion of temperature measurement, when this function is used, the ADCG1ULER upper/lower limit error register is set as if a lower-limit error has occurred. The upper/lower limit error bits of the ADCG1 must thus be cleared after temperature measurement.

CAUTION

When AN112 inputs a hold trigger of T&H group A or B during conversion, the virtual channel operation is suspended and the correct temperature measurement result cannot be obtained. Do not input a hold trigger of T&H group A or B during temperature measurement.

11.4.2 Temperature Error Notification

- TCN0 can use the ADCG1 upper-and-lower limit checking function to send a limit error signal to the ECM module when the temperature exceeds the specified upper or lower limit.
- The following settings are required to use Temperature error notification.
 1. Set the upper and lower limits of the temperature in the ADCG1ULLMTBR0 to 2 registers of ADCG1.
 2. Select the upper and lower limits table for scan group 4 in the ADCG1ULLMSR4 register of ADCG1.
 3. Set the ULIEI bit of the ADCG1SFTCR safety control register of ADCG1 to “1” to enable upper/lower limit error interrupts. Set the OWEIE, PEIE, and IDEIE bits to “0” to disable overwrite error interrupts, parity error interrupts, and ID error interrupts. Disable the ADCG1 error interrupt (INTADCG1ERR) of the interrupt controller (INTC).

11.4.3 Self-Diagnosis Function

TSN0 supports the insertion of faulty values. Setting the TSNSELFDIAG bit of the TSN0DIAG register to “1” forces the TSN0 module to generate an error. TSN0 can then diagnose its own operation by comparing the change result from the AN112 channel of ADCG1 with the expected value (0).

TSN0 is set up for self-diagnosis by the following register settings.

1. Set up A/D conversion for the temperature sensor channel. The procedure is the same as for normal operation. Refer to **Section 11.4.1, Temperature Measurement**.
2. Enable the temperature sensor. The procedure is the same as for normal operation. Refer to **Section 11.4.1, Temperature Measurement**.
3. Configure the self-diagnosis register before starting A/D conversion. That is, set the TSNSELFDIAG bit of the TSN0DIAG register to “1”. TSOOUT is fixed to the low level after this.
4. Start temperature measurement. The procedure is the same as for normal operation. Refer to **Section 11.4.1, Temperature Measurement**.
5. Check the result of measurement. The procedure is the same as for normal operation. Refer to **Section 11.4.1, Temperature Measurement**.

11.5 Calculating the Temperature

Use the following formula to convert results of temperature measurement into degrees Celsius.

$$T_{\text{current}} = \frac{150 - 25}{R_{150} - R_{25}} \times (R - R_{25}) + 25$$

$$R = \frac{A1VREFH}{5.0} \times R_{\text{current}}$$

R_{current}: Result of A/D conversion at the current temperature

T_{current}: Current temperature

R₂₅: Reference result of A/D conversion at 25 °C.

R₁₅₀: Reference result of A/D conversion at 150 °C.

R₂₅ and R₁₅₀ are stored in TSNREFD.

CAUTION

The values of R₂₅ and R₁₅₀ stored in TSNREFD are measured at A1VREFH = 5.0 V. Apply the formula after converting to the applicable voltage if A1VREFH has a different value.

Section 12 Clock Controller

The clock controller supplies clock pulses inside the chip and to the external devices. The clock controller consists of a main oscillator circuit (Main OSC), a high-speed internal oscillator (HS IntOSC), a Phase Locked Loop circuit (PLL), clock dividers and clock selectors.

12.1 Features

- Incorporates main oscillator circuit (Main OSC), which is used as a reference clock of the PLL.
- Incorporates PLL circuit to generate high speed internal clocks by multiplying the Main OSC input.
- Incorporates a high-speed internal oscillator (HS IntOSC).
- Generates clock pulses used inside the chip from HS IntOSC, Main OSC and PLL.
- A clock monitor (CLMA0 to CLMA3) is included. (See **Section 31, Functional Safety** for details.)
- The frequency divided clock set by the division circuit can be output from the EXTCLKnO (n = 0,1) pins. Furthermore, Main OSC, CLK_LSB, CLK_CPU, and CLK_IOSC can be selected with register settings.

12.1.1 External Input/Output Pins

Table 12.1 shows the pins related to the clock controller.

Table 12.1 Pins Related to the Clock Controller

| Pin Function Name | Direction | Function |
|-------------------|-----------|--|
| X1 | Input | Connects a Main OSC crystal oscillator |
| X2 | Output | Connects a Main OSC crystal oscillator |
| EXTCLK0O | Output | External clock out 0 |
| EXTCLK1O | Output | External clock out 1 |

12.2 Overview

12.2.1 Type of Clocks

Table 12.2 shows the list of clocks, **Table 12.3** shows the operation clocks of each functional module.

Table 12.2 List of Clocks

| Clock Name | Symbol | Clock Frequency (MHz) | Remarks |
|-----------------------------|----------|--|--|
| CPU clock | CLK_CPU | (CPU) 160 (Global RAM) 80 (1 wait) (CodeFlash) 80 (1 wait) | Clock source: PLL output |
| High speed peripheral clock | CLK_HSB | 80 | |
| Low speed peripheral clock | CLK_LSB | 40 | |
| Internal Oscillator clock | CLK_IOSC | 8 | High-speed internal oscillator(HS IntOSC) *1/2 |
| WDTA counter clock | WDTACKI | 8/0.25 | 1/1 or 1/32 of CLK_IOSC (Configured by FLASH option) |
| External clock out 0 | EXTCLK00 | 1/1 to 1/1023 of CLK_CPU, CLK_LSB, CLK_IOSC or Main OSC | Configured by CKSC2C and CLK2DIV. Maximum output frequency: 20 MHz |
| External clock out 1 | EXTCLK10 | 1/1 to 1/1023 of CLK_CPU, CLK_LSB, CLK_IOSC or Main OSC | Configured by CKSC3C and CLK3DIV. Maximum output frequency: 20 MHz |
| Main OSC clock | CLK_MOSC | 16 | |
| ADC clock | CLK_ADC | 40/20 | Configured by CKSC8C register |

Table 12.3 Clocks and Functional Modules

| Clock Name | Functional Module Name |
|-----------------------------|--|
| CPU clock | PE1 (Master, Checker, Local RAM, INTC1, etc.), Code Flash (1 wait), ERAM (1 wait), Global RAM (1 wait) |
| High speed peripheral clock | INTC2, DMAC, DTS, DCRA, PIC, CSIG, CSIH, SCI, FLXA, SENT, PSI5, RLIN, TAUD, TAUJ, TSG3, ENCA, TAPA, TPBA, OSTM |
| Low speed peripheral clock | Data Flash (from 4 cycle), RS-CANFD, Clock controller |
| Internal Oscillator clock | CVM (digital noise filter), CLMA (monitor clock), ECM (delay timer : CLK_IOSC/2) |
| WDTA counter clock | WDTA |
| ADC clock | ADCG, TSN |
| MainOSC clock | RS-CANFD (xin) |

12.2.2 Block Diagram

Figure 12.1 shows the block diagram of the clock controller.

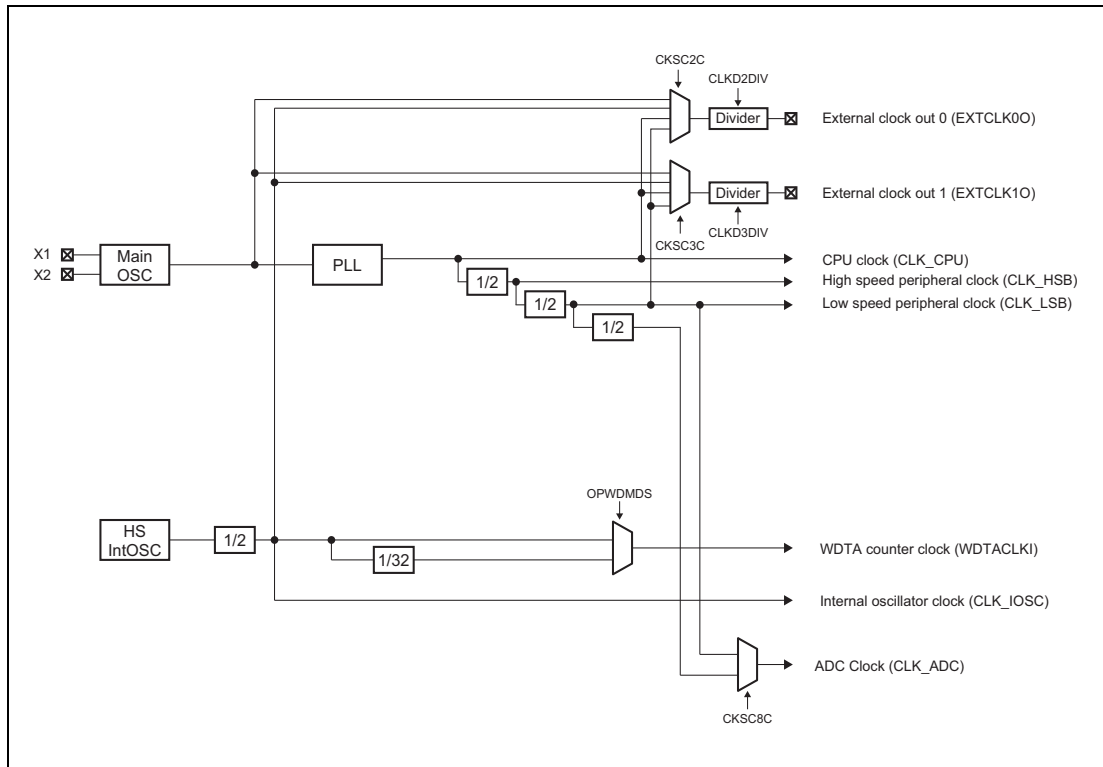


Figure 12.1 Block Diagram of Clock Controller

12.3 Register Description

12.3.1 Writing Protection to Registers

Registers can be protected from inadvertent write access due to erroneous program execution, etc. by configuration of the Slave Guards. For details, see **Section 31, Functional Safety**.

12.3.2 Register Overview

The Clock Controller is controlled and operated by the following registers:

Table 12.4 List of Registers

| Address | Register Name | Description | Access Width | Value after Reset |
|------------------------|---------------|-----------------------------------|--------------|------------------------|
| FFF8 8810 _H | CLKD2DIV | Clock divider 2 divisor register | 32 | 0000 0000 _H |
| FFF8 8814 _H | CLKD2STAT | Clock divider 2 status register | 32 | 0000 0002 _H |
| FFF8 8818 _H | CLKD3DIV | Clock divider 3 divisor register | 32 | 0000 0000 _H |
| FFF8 881C _H | CLKD3STAT | Clock divider 3 status register | 32 | 0000 0002 _H |
| FFF8 9080 _H | CKSC2C | Clock selector 2 control register | 32 | 0000 0004 _H |
| FFF8 9088 _H | CKSC2S | Clock selector 2 status register | 32 | 0000 0004 _H |
| FFF8 90C0 _H | CKSC3C | Clock selector 3 control register | 32 | 0000 0004 _H |
| FFF8 90C8 _H | CKSC3S | Clock selector 3 status register | 32 | 0000 0004 _H |
| FFF8 9110 _H | CKSC8C | Clock selector 8 control register | 32 | 0000 0002 _H |
| FFF8 9114 _H | CKSC8S | Clock selector 8 status register | 32 | 0000 0003 _H |

Table 12.5 Register Reset Conditions

| Register Name | Reset Condition | | | |
|---------------|-----------------|----------------|----------------|---------------------|
| | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| CLKD2DIV | √ | √ | √ | √ |
| CLKD2STAT | √ | √ | √ | √ |
| CLKD3DIV | √ | √ | √ | √ |
| CLKD3STAT | √ | √ | √ | √ |
| CKSC2C | √ | √ | √ | √ |
| CKSC2S | √ | √ | √ | √ |
| CKSC3C | √ | √ | √ | √ |
| CKSC3S | √ | √ | √ | √ |
| CKSC8C | √ | √ | √ | √ |
| CKSC8S | √ | √ | √ | √ |

12.3.3 CLKD2DIV — Clock Divider 2 Divisor Register

This register controls the division ratio of the external clock output 0 (EXTCLK00).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read/write in 32-bit units.

Address: FFF8 8810_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | CLKD2DIV[9:0] | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 12.6 CLKD2DIV Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 to 0 | CLKD2DIV | Division Ratio 0: EXTCLK00 is stopped (low level). 1 to 1023: The source for the EXTCLK00 is divided by the CLKD2DIV. |

CAUTION

Set the frequency of EXTCLK00 to a value less than 20 MHz.

Do not write to the CLKD2DIV register while the setting of the CLKD2SYNC bit is 0.

12.3.4 CLKD2STAT — Clock Divider 2 Status Register

This register indicates the status of the external clock output 0 (EXTCLK00).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read in 32-bit units.

Address: FFF8 8814_H

Value after reset: 0000 0002_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|-----------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLKD2 SYNC | CLKD2 CLKACT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 12.7 CLKD2STAT Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | CLKD2SYNC | Divider Clock Synchronized 0: EXTCLK00 output does not correspond to the division ratio setting in CLKD2DIV. 1: EXTCLK00 output corresponds to the division ratio setting in CLKD2DIV. |
| 0 | CLKD2CLKACT | Divider Clock Active 0: EXTCLK00 is inactive (EXTCLK00 = low level). 1: EXTCLK00 is active. |

12.3.5 CLKD3DIV — Clock Divider 3 Divisor Register

This register controls the division ratio of the external clock output 1 (EXTCLK1O).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read/write in 32-bit units.

Address: FFF8 8818_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | — | — | — | — | — | — | CLKD3DIV[9:0] | | | | | | | | | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

Table 12.8 CLKD3DIV Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 to 0 | CLKD3DIV | Division Ratio 0: EXTCLK1O is stopped (low level). 1 to 1023: The source for the EXTCLK1O is divided by the CLKD3DIV. |

CAUTION

Set the frequency of EXTCLK1O to a value less than 20 MHz.

Do not write to the CLKD3DIV register while the setting of the CLKD3SYNC bit is 0.

12.3.6 CLKD3STAT — Clock Divider 3 Status Register

This register indicates the status of the the external clock output 1 (EXTCLK1O).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read in 32-bit units.

Address: FFF8 881C_H

Value after reset: 0000 0002_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|-----------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLKD3 SYNC | CLKD3 CLKACT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 12.9 CLKD3STAT Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | CLKD3SYNC | Divider Clock Synchronized 0: EXTCLK1O output does not correspond to the division ratio setting in CLKD3DIV. 1: EXTCLK1O output corresponds to the division ratio setting in CLKD3DIV. |
| 0 | CLKD3CLKACT | Divider Clock Active 0: EXTCLK1O is inactive (EXTCLK1O = low level). 1: EXTCLK1O is active. |

12.3.7 CKSC2C — Clock Selector 2 Control Register

This register selects the clock which drives the external clock output 0 (EXTCLK00).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read/write in 32-bit units.

Address: FFF8 9080_H

Value after reset: 0000 0004_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CKSC2 | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 12.10 CKSC2C Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | CKSC2 | Clock Source Control These bits select the ID of a clock source for the EXTCLK00. ID = 3 _H : Main OSC ID = 4 _H : CLK_LSB ID = 5 _H : CLK_CPU ID = 6 _H : CLK_IOOSC Other than above: setting prohibited. |

CAUTION

The CKSC2C register must not be modified when the CLKD2DIV register is not 0000 0000_H or the CLKD2STAT register is not 0000 0002_H.

While CLKD2DIV = 0000 0000_H and CLKD2STAT = 0000 0002_H, EXTCLK00 output is stopped. Change the value of the register in this situation.

12.3.8 CKSC2S — Clock Selector 2 Status Register

The register indicates the state of a clock source for the external output 0 (EXTCLK00).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read in 32-bit units.

Address: FFF8 9088_H

Value after reset: 0000 0004_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CLKACT2 | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 12.11 CKSC2S Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. |
| 2 to 0 | CLKACT2 | Clock Source Status Indicates the ID of the selected clock source of EXTCLK00. The value indicated by these bits shows the actual ID that is currently selected or currently waiting to be changed to another ID. ID = 3 _H : Main OSC ID = 4 _H : CLK_LSB ID = 5 _H : CLK_CPU ID = 6 _H : CLK_IOSC |

12.3.9 CKSC3C — Clock Selector 3 Control Register

This register selects the clock which drives the external clock output 1 (EXTCLK1O).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read/write in 32-bit units.

Address: FFF8 90C0_H

Value after reset: 0000 0004_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CKSC3 | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 12.12 CKSC3C Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | CKSC3 | Clock Source Control These bits select the ID of a clock source for the EXTCLK1O. ID = 3 _H : Main OSC ID = 4 _H : CLK_LSB ID = 5 _H : CLK_CPU ID = 6 _H : CLK_IOSC Other than above: setting prohibited. |

CAUTION

The CKSC3C register must not be modified when the CLKD3DIV register is not 0000 0000_H or the CLKD3STAT register is not 0000 0002_H.

While CLKD3DIV = 0000 0000_H and CLKD3STAT = 0000 0002_H, EXTCLK1O output is stopped. Change the value of the register in this situation.

12.3.10 CKSC3S — Clock Selector 3 Status Register

The register indicates the state of a clock source for the external output 1 (EXTCLK1O).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read in 32-bit units.

Address: FFF8 90C8_H

Value after reset: 0000 0004_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CLKACT3 | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 12.13 CKSC3S Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | When read, the value after reset is read. |
| 2 to 0 | CLKACT3 | Clock Source Status Indicates the ID of the selected clock source of EXTCLK1O. The value indicated by these bits shows the actual ID that is currently selected or currently waiting to be changed to another ID. ID = 3 _H : Main OSC ID = 4 _H : CLK_LSB ID = 5 _H : CLK_CPU ID = 6 _H : CLK_IOSOC |

12.3.11 CKSC8C — Clock Selector 8 Control Register

This register selects the clock of ADCG.

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read in 32-bit units.

Address: FFF8 9110_H

Value after reset: 0000 0002_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------|----------------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CKSCI D8[1] | CKSCI D8[0] | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R |

Table 12.14 CKSC8C Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2, 1 | CKSCID8[1:0] | Clock Source Control These bits select the ID of a clock source for the ADCG. ID = 00 _B : setting prohibited ID = 01 _B : CLK_LSB (40 MHz, default) ID = 10 _B : CLK_LSB /2 (20 MHz) ID = 11 _B : setting prohibited |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

CAUTION

Do not change the setting of the clock selection bits while the A/D converter is in use.

12.3.12 CKSC8S — Clock Selector 8 Status Register

This register indicates the status of a clock source of ADBG.

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: This register can be read in 32-bit units.

Address: FFF8 9114_H

Value after reset: 0000 0003_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|--------------|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CLKSELID8[1] | CLKSELID8[0] | CLKACT8 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 12.15 CKSC8S Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. |
| 2, 1 | CLKSELID8[1:0] | Clock Source Status Indicates the ID of the selected clock source of ADBG. The value indicated by these bits shows the actual ID that is currently selected or currently waiting to be changed to another ID. ID = 00 _B : setting prohibited ID = 01 _B : CLK_LSB (40 MHz, default) ID = 10 _B : CLK_LSB /2 (20 MHz) ID = 11 _B : setting prohibited |
| 0 | CLKACT8 | Clock Active Indicates that the clock of the corresponding clock domain is active and synchronized with the selected clock. 0: Non-active 1: Active |

12.4 Operation

12.4.1 External Clock Output

The device provides 2 continuous external clock outputs that can be used as clock supply for external circuits. The clock selection can be configured by software. For details, see **Section 12.3.3, CLKD2DIV — Clock Divider 2 Divisor Register, Section 12.3.4, CLKD2STAT — Clock Divider 2 Status Register, Section 12.3.5, CLKD3DIV — Clock Divider 3 Divisor Register, Section 12.3.6, CLKD3STAT — Clock Divider 3 Status Register, Section 12.3.7, CKSC2C — Clock Selector 2 Control Register, Section 12.3.8, CKSC2S — Clock Selector 2 Status Register, Section 12.3.9, CKSC3C — Clock Selector 3 Control Register, and Section 12.3.10, CKSC3S — Clock Selector 3 Status Register.**

- Clock signals can be output from the EXTCLKnO (n = 0, 1) pins by using the clock output function.
- Output clock frequencies can be divided by the division circuit according to the register settings.
- Main OSC, CLK_LSB, CLK_CPU, and CLK_IOSC can be selected as the clock source by using register settings.

To produce an output on the EXTCLKnO pin at a specified value, follow the procedure below to make the settings to select the source clock and division ratio.

- Selecting the source clock
 1. Check the following to confirm that EXTCLKnO output is stopped.
 - $CLKDnDIV = 0000\ 0000_H$ and
 - $CLKDnSTAT = 0000\ 0002_H$
($CLKDnSTAT.CLKDnCLKACT = 0$, $CLKDnSTAT.CLKDnSYNC = 1$)
 2. Set $CKSCnC.CKSCn[2:0]$ to select the clock signal.
 3. Read the $CKSCnS$ register to confirm that the selected clock signal is active.
 - Confirm that the value of $CKSCnS.CLKACTn[2:0] =$ the value of $CKSCnC.CKSCn[2:0]$.
- Selecting the clock divisor
 1. Confirm that $CLKDnSTAT.CLKDnSYNC = 1$.
 2. Set $CLKDnDIV$ to select the EXTCLKnO output division ratio.
 3. Confirm that $CLKDnSTAT.CLKDnSYNC = 1$.
- Changing the setting of the clock selection bits
 1. Follow the above procedure for selecting the clock divisor to set $CLKDnDIV = 0000\ 0000_H$ (stopping output of clock signals).
 2. Follow the above procedure for selecting the source clock to set $CKSCnC.CKSCn[2:0]$ to select the changed clock signal.

12.5 Usage Notes

12.5.1 How to Connect a Crystal Oscillator

Figure 12.2 shows how to connect a Crystal Oscillator. When the resonator recommended by the company is used (contact us for details), external components such as a load capacitor and a damping resistor are not necessarily required for oscillation. Please evaluate the oscillation in your environment before use.

As shown in **Figure 12.2**, do not cross any other signal lines over the signal lines to the X1 and X2 pins. Induction may inhibit proper oscillation.

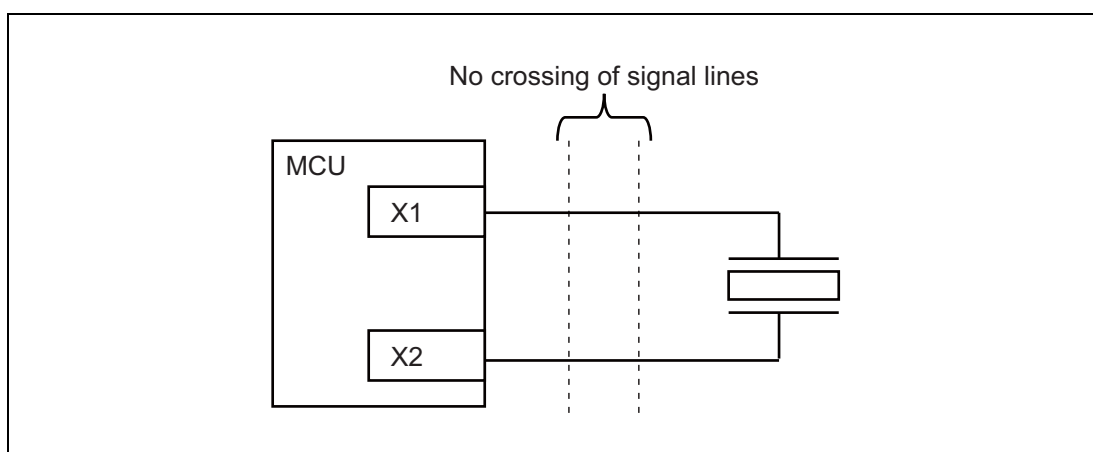


Figure 12.2 Connection Example of Crystal Oscillator

Section 13 Clocked Serial Interface G (CSIG)

The Clocked Serial Interface (CSI) module is a serial data link also well known as SPI, serial peripheral interface.

The CSI bus is a synchronous serial data link standard normally operating in full duplex mode via a 3-wire serial I/O.

Communication is set up in master/slave mode where the master initiates the data frame and provides the clock.

This microcontroller has one unit of CSIG.

13.1 Features of RH850/P1M-E CSIG

13.1.1 Number of Units

This microcontroller has the following number of CSIG units.

Each CSIG unit has one channel interface. "Number of channels" is used with the same meaning as "number of units" in this section.

Table 13.1 Number of Units

| Product Name | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|------------------------------|-------------------------|
| Number of Units | 1 | |
| Name | CSIG _n (n = 0) | |

Note: The channel names are same as those of the corresponding units.

Table 13.2 Index

| Index | Description |
|-------|--|
| n | Throughout this section, the individual CSIG units are identified by the index "n" (n = 0): for example, CSIG _n CTL0 is the CSIG _n control register 0. |

13.1.2 Register Base Address

CSIG base addresses are listed in the following table.

CSIG register addresses are given as offsets from the base addresses throughout the section.

Table 13.3 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <CSIG0_base> | FFD8 A000 _H |

13.1.3 Clock Supply

Clock supply by and to CSIG is listed in the following table.

Table 13.4 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|-------------------------------------|
| CSIGn | PCLK | High-speed peripheral clock CLK_HSB |

13.1.4 Interrupt Requests

CSIG interrupt requests are listed in the following table.

Table 13.5 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|------------------------------------|------------------|------------------------|
| CSIG0 | | | |
| INTCSIG0IC | CSIG transmit status interrupt | 175 | 86 |
| INTCSIG0IR | CSIG receive status interrupt | 176 | 85 |
| INTCSIG0IRE | CSIG communication error interrupt | 174 | — |

13.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

13.1.6 External Input/Output Signals

External input/output signals of CSIG are listed below.

Table 13.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|------------------|-----|----------------------------|----------------------------------|
| CSIG0 | | | |
| CSIGTSCK | I | Serial clock signal input | CSIG0SCI |
| | O | Serial clock signal output | CSIG0SCO |
| CSIGTSI | I | Serial data input signal | CSIG0SI |
| CSIGTSO | O | Serial data output signal | CSIG0SO |
| CSIGTRYI | I | Ready/busy input signal | CSIG0RYI |
| CSIGTRYO | O | Ready/busy output signal | CSIG0RYO |

13.1.7 Data Consistency Check

The following table lists the port pins on which CSIGNSO pin functions are multiplexed and whether or not the CSIGNSO pin functions support data consistency checking. See **Section 13.5.10, Error Detection** for details on data consistency checking.

Table 13.7 Data Consistency Checking and Target Port Pins

| Unit Signal Name | Port Pin Name | Alternative Function | Data Consistency Checking |
|------------------|---------------|----------------------|---------------------------|
| CSIG0 | | | |
| CSIGTSO | P3_1*1 | ALT-OUT6 | Supported |
| | P5_1 | ALT-OUT1 | Supported |

Note 1. Available in devices with 144-pin.

13.1.8 Combinations of Pins and Ports

Combinations of CSIG pins and ports are listed in the following table.

Table 13.8 Combinations of Pins and Ports

| Function | Pin Name | Port Name | | |
|----------|-----------------------|-----------|---------|---------|
| | | Group 1 | Group 2 | Group 3 |
| CSIG0 | CSIG0RYI | P3_5 | — | — |
| | CSIG0RYO | P3_4 | — | — |
| | CSIG0SCI/ CSIG0SCO | P3_2*1 | P5_2*1 | P5_4 |
| | CSIG0SI | P3_0*1 | P5_0 | P5_0 |
| | CSIG0SO | P3_1*1 | P5_1 | P5_1 |

Note 1. Available in devices with 144-pin.

13.2 Overview

13.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode and slave mode
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode. In slave mode, transfer clock frequency is determined by input clock.
- Maximum transmission speed:
 - in master mode: 8 MHz
 - in slave mode: 6.6 MHz
- Clock phase and data phase are selectable.
- Data transfer with MSB first or data transfer with LSB first is selectable.
- Transfer data length selectable from 7 to 16 bits in 1-bit units
- EDL (Extended Data Length) function for transferring data with more than 16 bits is incorporated.
- Three selectable transfer modes:
 - transmit-only mode
 - receive-only mode
 - transmit/receive mode
- Handshake function is included.
- Built-in error detection (data consistency check, parity, overrun)
- Three different interrupt request signals (INTCSIG0IC, INTCSIG0IR, INTCSIG0IRE)
- Built-in LBM (Loop Back Mode) function for self-test

13.2.2 Functional Description

The CSIG uses three signals for communication:

- Transmission clock CSIGTCK (output in master mode, input in slave mode)
- Serial data output signal CSIGTSO
- Serial data input signal CSIGTSI

CSIGNCTL2 is used to select the mode in which the CSIG should be operated: master mode or slave mode.

Other signals are available for external control and monitoring.

- CSIGTRYO: Ready/busy output signal (handshake signal)
- CSIGTRYI: Ready/busy input signal (handshake signal)

Data transmission is bit-wise and serial, and synchronous to the transmission clock.

The following table shows the most important registers for setting up CSIG.

Table 13.9 Main Registers of CSIG

| Register | Function |
|-----------|---|
| CSIGNCTL0 | Enables/disables operating clock, data transmission, and data reception |
| CSIGNCTL1 | Controls options such as interrupt timing, extended data length, data consistency check, loop-back mode, handshake, etc. |
| CSIGNCTL2 | Selects master or slave mode and also selects the transfer clock frequency of the on-chip baud rate generator (BRG) in master mode. |
| CSIGNCFG0 | Configures the communication protocol |

13.2.3 Block Diagram

The following block diagram shows the main components of the CSIG.

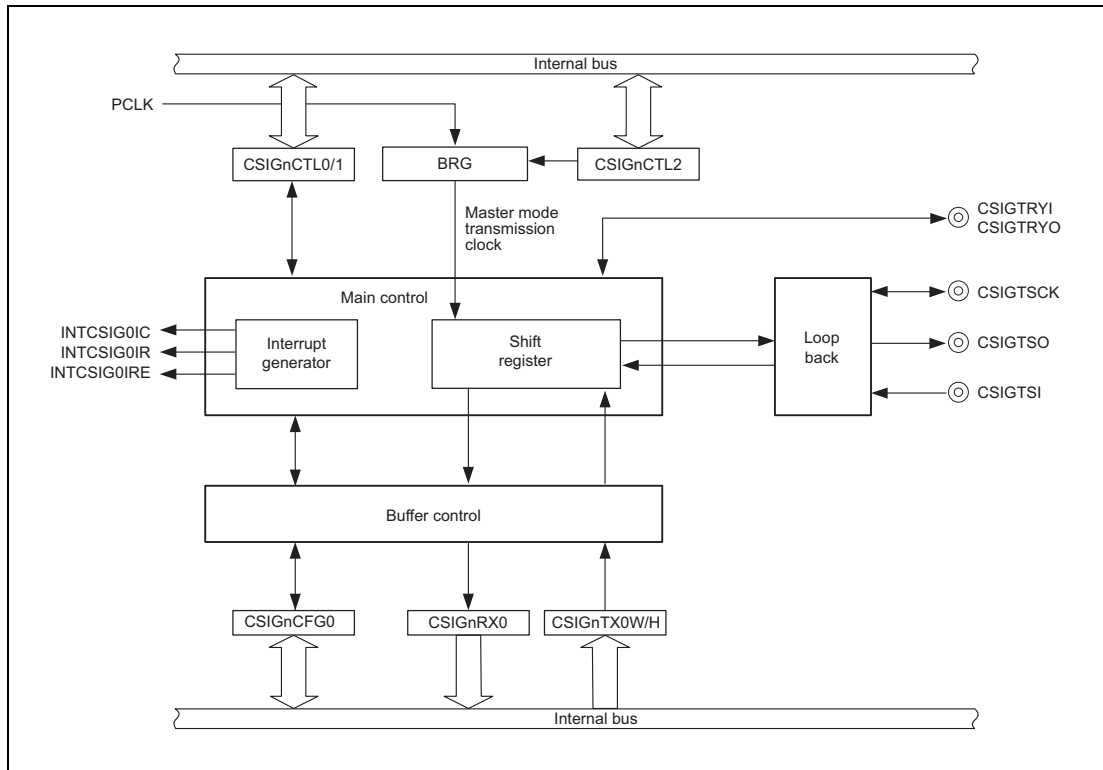


Figure 13.1 CSIG Block Diagram

In master mode, the transmission clock CSIGTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

13.3 Registers

13.3.1 List of Registers

CSIG registers are listed in the following table.

The base address <CSIGn_base> of the CSIGn is defined in **Section 13.1.2, Register Base Address**.

Table 13.10 Registers

| Module | Register | Symbol | Address |
|--------|--|------------|----------------------------------|
| CSIGn | CSIGn control register 0 | CSIGnCTL0 | <CSIGn_base> + 0000 _H |
| CSIGn | CSIGn control register 1 | CSIGnCTL1 | <CSIGn_base> + 0010 _H |
| CSIGn | CSIGn control register 2 | CSIGnCTL2 | <CSIGn_base> + 0014 _H |
| CSIGn | CSIGn status register 0 | CSIGnSTR0 | <CSIGn_base> + 0004 _H |
| CSIGn | CSIGn status clear register 0 | CSIGnSTCR0 | <CSIGn_base> + 0008 _H |
| CSIGn | CSIGn Rx-only mode control register 0 | CSIGnBCTL0 | <CSIGn_base> + 1000 _H |
| CSIGn | CSIGn configuration register 0 | CSIGnCFG0 | <CSIGn_base> + 1010 _H |
| CSIGn | CSIGn transmission register 0 for word access | CSIGnTX0W | <CSIGn_base> + 1004 _H |
| CSIGn | CSIGn transmission register 0 for half word access | CSIGnTX0H | <CSIGn_base> + 1008 _H |
| CSIGn | CSIGn reception register 0 | CSIGnRX0 | <CSIGn_base> + 100C _H |

13.3.2 CSIGNCTL0 — CSIGN Control Register 0

This register controls the operation clock and enables or disables transmission/reception.

Access: This register can be read/written in 8-bit units.

Address: <CSIGN_base> + 0000_H

Value after reset: 00_H This register is initialized by a reset from any source.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|----------|----------|---|---|---|---|----------|
| | CSIGNPWR | CSIGNTXE | CSIGNRXE | — | — | — | — | CSIGNMBS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R/W |

Table 13.11 CSIGNCTL0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | CSIGNPWR | Controls operation clock: 0: Stop operation clock 1: Supplies operation clock Clearing CSIGNPWR to 0 resets the internal circuits, stops operation, and sets the CSIG to standby state. Clock supply to internal circuits stops. If CSIGNPWR is cleared during communication, ongoing communication is aborted. In that case, communication settings must be made from the beginning. |
| 6 | CSIGNTXE | Enables/disables transmission: 0: Transmission disabled 1: Transmission enabled |
| 5 | CSIGNRXE | Enables/disables reception: 0: Reception disabled 1: Reception enabled |
| 4 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | CSIGNMBS | To use CSIG, always write 1 to this bit. (The value after reset is "0".) |

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.3 CSIGNCTL1 — CSIGN Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, and handshake function.

Access: This register can be read/written in 32-bit units.

Address: <CSIGN_base> + 0010_H

Value after reset: 0000 0000_H This register is initialized by a reset from any source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|-----------|----|----------|----|----------|----------|----------|-----------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CSIGNCKR | CSIGNSLIT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | CSIGNEDLE | — | CSIGNDCS | — | CSIGNLBM | CSIGNSIT | CSIGNHSE | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R | R/W | R | R/W | R/W | R/W | R |

Table 13.12 CSIGNCTL1 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17 | CSIGNCKR | CSIGTSCK clock inverting function 0: The default level of CSIGTSCK is high level. 1: The default level of CSIGTSCK is low level. The CSIGNCKR bit is used in combination with the CSIGNCFG0.CSIGNDAP bit. For details, refer to Section 13.3.8, CSIGNCFG0 — CSIGN Configuration Register 0 . |
| 16 | CSIGNSLIT | Selects the timing of interrupt INTCSIG0IC: 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt generation when CSIGNTX0W/H is ready to store the (next data.). For details, see Section 13.4.2, INTCSIG0IC (Communication Status Interrupt) |
| 15 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 | CSIGNEDLE | Enables/disables extended data length (EDL) mode: 0: Extended data length mode disabled 1: Extended data length mode enabled For details, refer to Section 13.5.5.2, Data Length Selection with Extended Data Length . |
| 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | CSIGNDCS | Enables/disables data consistency check: 0: Data consistency check disabled 1: Data consistency check enabled For details, refer to Section 13.5.10.1, Data Consistency Check . |
| 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | CSIGNLBM | Controls loop-back mode (LBM): 0: Loop-back mode deactivated 1: Loop-back mode activated Loop-back mode can be set only in master mode. Set this bit to 0 in slave mode. For details, refer to Section 13.5.9, Loop-Back Mode . |

Table 13.12 CSIGnCTL1 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 2 | CSIGnSIT | Selects interrupt delay mode: 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, see Section 13.4.1, Interrupt Delay . |
| 1 | CSIGnHSE | Enables or disables the handshake function: 0: Handshake function disabled 1: Handshake function enabled For details refer to Section 13.5.8, Handshake Function . |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers**.

13.3.4 CSIGNCTL2 — CSIGN Control Register 2

This register selects the communication clock.

Access: This register can be read/written in 16-bit units.

Address: <CSIGN_base> + 0014_H

Value after reset: E000_H This register is initialized by a reset from any source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|-----|----|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | CSIGNPRS[2:0] | | | — | CSIGNBRS[11:0] | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 13.13 CSIGNCTL2 Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|--------------------------------------|---|--|--------------------------------------|-----------|---------------------------|---|---------------------------------|---|---------------------------------|---|---------------------------------|---|---------------------------------|-----|-----|------|------------------------------------|---|---|---|------------------------|---|---|---|-------------------------|---|---|---|-------------------------|---|---|---|-------------------------|---|---|---|--|
| 15 to 13 | CSIGNPRS [2:0] | Selects the value m of the prescaler: <table border="1" data-bbox="678 846 1417 1205"> <thead> <tr> <th>CSIGNPRS2</th> <th>CSIGNPRS1</th> <th>CSIGNPRS0</th> <th>Prescaler output (PRSOUT)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>PCLK (master mode)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>PCLK / 2 (master mode)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>PCLK / 4 (master mode)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>PCLK / 8 (master mode)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>PCLK / 16 (master mode)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>PCLK / 32 (master mode)</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>PCLK / 64 (master mode)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>External clock via CSIGTSCK (slave mode)</td> </tr> </tbody> </table> | CSIGNPRS2 | CSIGNPRS1 | CSIGNPRS0 | Prescaler output (PRSOUT) | 0 | 0 | 0 | PCLK (master mode) | 0 | 0 | 1 | PCLK / 2 (master mode) | 0 | 1 | 0 | PCLK / 4 (master mode) | 0 | 1 | 1 | PCLK / 8 (master mode) | 1 | 0 | 0 | PCLK / 16 (master mode) | 1 | 0 | 1 | PCLK / 32 (master mode) | 1 | 1 | 0 | PCLK / 64 (master mode) | 1 | 1 | 1 | External clock via CSIGTSCK (slave mode) |
| CSIGNPRS2 | CSIGNPRS1 | CSIGNPRS0 | Prescaler output (PRSOUT) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | PCLK (master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | PCLK / 2 (master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | PCLK / 4 (master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | PCLK / 8 (master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | PCLK / 16 (master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | PCLK / 32 (master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | PCLK / 64 (master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | External clock via CSIGTSCK (slave mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 to 0 | CSIGNBRS [11:0] | Selects the transfer clock frequency. Settings of the CSIGNBRS[11:0] bits are valid only in master mode. They are ignored in slave mode. <table border="1" data-bbox="678 1377 1417 1722"> <thead> <tr> <th>CSIGNBRS [11:0]</th> <th>Transfer Clock Frequency at CSIGTSCK</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>BRG is stopped</td> </tr> <tr> <td>1</td> <td>PCLK / (2^α × 1 × 2)</td> </tr> <tr> <td>2</td> <td>PCLK / (2^α × 2 × 2)</td> </tr> <tr> <td>3</td> <td>PCLK / (2^α × 3 × 2)</td> </tr> <tr> <td>4</td> <td>PCLK / (2^α × 4 × 2)</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>4095</td> <td>PCLK / (2^α × 4095 × 2)</td> </tr> </tbody> </table> | CSIGNBRS [11:0] | Transfer Clock Frequency at CSIGTSCK | 0 | BRG is stopped | 1 | PCLK / (2 ^α × 1 × 2) | 2 | PCLK / (2 ^α × 2 × 2) | 3 | PCLK / (2 ^α × 3 × 2) | 4 | PCLK / (2 ^α × 4 × 2) | ... | ... | 4095 | PCLK / (2 ^α × 4095 × 2) | | | | | | | | | | | | | | | | | | | | |
| CSIGNBRS [11:0] | Transfer Clock Frequency at CSIGTSCK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | BRG is stopped | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | PCLK / (2 ^α × 1 × 2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | PCLK / (2 ^α × 2 × 2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | PCLK / (2 ^α × 3 × 2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | PCLK / (2 ^α × 4 × 2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4095 | PCLK / (2 ^α × 4095 × 2) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Note: α = 0 to 6 (value set by CSIGNPRS[2:0])

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.5 CSIGNSTR0 — CSIGN Status Register 0

This register indicates the status of the CSIG.

Access: This register can only be read in 32-bit units.

Address: <CSIGN_base> + 0004_H

Value after reset: 0000 0010_H This register is initialized by a reset from any source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|--------------|----|----|----|--------------|----|-------------|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | CSIGN TSF | — | — | — | CSIGN DCE | — | CSIGN PE | CSIGN OVE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 13.14 CSIGNSTR0 Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | |
|--------------|-------------------------------|---|-------------|---------------|-----------------|--------------|------------------------------|---|------------|--------------|-------------------------------|--|------------|---------------|-----------------|--------------|------------------------------|---|------------|--------------|-----------------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. | | | | | | | | | | | | | | | | | | | | |
| 7 | CSIGNTSF | Transfer Status Flag 0: Idle state 1: Communication is in progress or being prepared The timing to set or clear this bit is as follows: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Master mode</th><th>Timing to Set</th><th>Timing to Clear</th></tr> </thead> <tbody> <tr> <td>Tx only mode</td><td rowspan="2">Writing to transmit register</td><td rowspan="2">Within a half clock cycle from the last serial clock edge</td></tr> <tr> <td>Tx/Rx mode</td></tr> <tr> <td>Rx only mode</td><td>Reading from receive register</td><td></td></tr> </tbody> </table> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Slave mode</th><th>Timing to Set</th><th>Timing to Clear</th></tr> </thead> <tbody> <tr> <td>Tx only mode</td><td rowspan="2">Writing to transmit register</td><td rowspan="2">Within a half clock cycle from the last serial clock edge</td></tr> <tr> <td>Tx/Rx mode</td></tr> <tr> <td>Rx only mode</td><td>CSIGNTSCK input</td><td></td></tr> </tbody> </table> | Master mode | Timing to Set | Timing to Clear | Tx only mode | Writing to transmit register | Within a half clock cycle from the last serial clock edge | Tx/Rx mode | Rx only mode | Reading from receive register | | Slave mode | Timing to Set | Timing to Clear | Tx only mode | Writing to transmit register | Within a half clock cycle from the last serial clock edge | Tx/Rx mode | Rx only mode | CSIGNTSCK input | |
| Master mode | Timing to Set | Timing to Clear | | | | | | | | | | | | | | | | | | | | |
| Tx only mode | Writing to transmit register | Within a half clock cycle from the last serial clock edge | | | | | | | | | | | | | | | | | | | | |
| Tx/Rx mode | | | | | | | | | | | | | | | | | | | | | | |
| Rx only mode | Reading from receive register | | | | | | | | | | | | | | | | | | | | | |
| Slave mode | Timing to Set | Timing to Clear | | | | | | | | | | | | | | | | | | | | |
| Tx only mode | Writing to transmit register | Within a half clock cycle from the last serial clock edge | | | | | | | | | | | | | | | | | | | | |
| Tx/Rx mode | | | | | | | | | | | | | | | | | | | | | | |
| Rx only mode | CSIGNTSCK input | | | | | | | | | | | | | | | | | | | | | |
| 6 to 4 | Reserved | When read, the value after reset is read. | | | | | | | | | | | | | | | | | | | | |
| 3 | CSIGNDCE | Data Consistency Check Error Flag 0: No data consistency check error detected 1: Data consistency check error detected This bit is cleared by writing 1 to CSIGNSTR0.CSIGNDCEC. However, if setting this bit to 1 due to detection of a data consistency check error and clearing this bit to 0 by CSIGNDCEC in CSIGNSTR0 take place at the same time, setting this bit to 1 due to detection of a data consistency check error takes precedence. This bit is initialized when CSIGNPWR in CSIGNCTL0 changes from 0 to 1 or from 1 to 0. | | | | | | | | | | | | | | | | | | | | |
| 2 | Reserved | When read, the value after reset is read. | | | | | | | | | | | | | | | | | | | | |

Table 13.14 CSIGnSTR0 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 1 | CSIGnPE | <p>Parity Error Flag</p> <p>0: No parity error detected 1: Parity error detected</p> <p>This bit is cleared by writing 1 to CSIGnPEC in CSIGnSTCR0. However, if setting this bit to 1 due to detection of a parity error and clearing this bit to 0 by CSIGnPEC in CSIGnSTCR0 take place at the same time, setting this bit to 1 due to detection of a parity error takes precedence. This bit is initialized when CSIGnPWR in CSIGnCTL0 changes from 0 to 1 or from 1 to 0.</p> |
| 0 | CSIGnOVE | <p>Overrun Error Flag</p> <p>0: No overrun error detected 1: Overrun error detected</p> <p>This bit is cleared by writing 1 to CSIGnOVEC in CSIGnSTCR0. However, if setting this bit to 1 due to detection of an overrun error and clearing this bit to 0 by CSIGnOVEC in CSIGnSTCR0 take place at the same time, setting this bit to 1 due to detection of an overrun error takes precedence. This bit is initialized when CSIGnPWR in CSIGnCTL0 changes from 0 to 1 or from 1 to 0.</p> |

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.6 CSIGNSTCR0 — CSIGN Status Clear Register 0

This register clears the status flags of the CSIGNSTR0 status register.

Access: This register can be read/written in 16-bit units.
When read, the value 0000_H is always returned.

Address: <CSIGN_base> + 0008_H

Value after reset: 0000_H This register is initialized by a reset from any source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---------------|---|--------------|---------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | CSIGN DCEC | — | CSIGN PEC | CSIGN OVEC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R | R/W | R/W |

Table 13.15 CSIGNSTCR0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 15 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | CSIGNDCEC | This bit controls the data consistency check error flag clear command. 0: No operation. 1: Clears the data consistency check error flag (CSIGNSTR0.CSIGNDCE) Read value is always 0. |
| 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | CSIGNPEC | This bit controls the parity error flag clear command. 0: No operation. Read value is always 0. 1: Clears the parity error flag (CSIGNSTR0.CSIGNPE) Read value is always 0. |
| 0 | CSIGNOVEC | This bit controls the overrun error flag clear command. 0: No operation. Read value is always 0. 1: Clears the overrun error flag (CSIGNSTR0.CSIGNOVE) Read value is always 0. |

13.3.7 CSIGNBCTL0 — CSIGN Rx-only Mode Control Register 0

This register enables/disables the data transfer in Rx-only mode.

Access: This register can be read/written in 8-bit units.

Address: <CSIGN_base> + 1000_H

Value after reset: 01_H This register is initialized by a reset from any source.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|----------|
| | — | — | — | — | — | — | — | CSIGNSCE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 13.16 CSIGNBCTL0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | CSIGNSCE | Disables/enables next data reception start by reading CSIGNRX0 0: Next reception disabled 1: Next reception enabled For details refer to Section 13.5.4.2, Receive-Only Mode. |

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.8 CSIGNCFG0 — CSIGN Configuration Register 0

This register configures the communication protocol options such as data length, parity, transfer direction, clock phase, and data phase.

Access: This register can be read/written in 32-bit units.

Address: <CSIGN_base> + 1010_H

Value after reset: 0000 0000_H This register is initialized by a reset from any source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|--------------|-----|---------------|-----|-----|-----|----|----|----|----|----|-----------|----|-----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | CSIGNPS[1:0] | | CSIGNDLS[3:0] | | | | — | — | — | — | — | CSIGN DIR | — | CSIGN DAP |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R/W | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 13.17 CSIGNCFG0 Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | |
|--------------|----------------|---|---------------------------------------|-----------|--------------|-----------|---|---|-----------------------|--------------------|---|---|---------------------------|---------------------------------------|---|---|----------------|----------------------------|---|---|-----------------|-----------------------------|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | |
| 29, 28 | CSIGNPS[1:0] | Specifies parity. <table border="1" data-bbox="678 1146 1417 1355"> <thead> <tr> <th>CSIGN PS1</th> <th>CSIGN PS0</th> <th>Transmission</th> <th>Reception</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No parity transmitted</td> <td>No parity expected</td> </tr> <tr> <td>0</td> <td>1</td> <td>Add parity bit fixed at 0</td> <td>Parity bit is expected but not judged</td> </tr> <tr> <td>1</td> <td>0</td> <td>Add odd parity</td> <td>Odd parity bit is expected</td> </tr> <tr> <td>1</td> <td>1</td> <td>Add even parity</td> <td>Even parity bit is expected</td> </tr> </tbody> </table> | CSIGN PS1 | CSIGN PS0 | Transmission | Reception | 0 | 0 | No parity transmitted | No parity expected | 0 | 1 | Add parity bit fixed at 0 | Parity bit is expected but not judged | 1 | 0 | Add odd parity | Odd parity bit is expected | 1 | 1 | Add even parity | Even parity bit is expected |
| CSIGN PS1 | CSIGN PS0 | Transmission | Reception | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | No parity transmitted | No parity expected | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Add parity bit fixed at 0 | Parity bit is expected but not judged | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Add odd parity | Odd parity bit is expected | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Add even parity | Even parity bit is expected | | | | | | | | | | | | | | | | | | | |
| 27 to 24 | CSIGNDLS [3:0] | Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits CAUTION Do not set bits CSIGNCFG0.CSIGNDLS[3:0] to values 1 to 6 when the extended data length function is disabled with bit CSIGNCTL1.CSIGNEDLE set to 0. Furthermore, Transmitting two consecutive data with a data length of less than 7 bits is prohibited. | | | | | | | | | | | | | | | | | | | | |
| 23 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | |
| 18 | CSIGNDIR | Selects the serial data direction. 0: Data is transmitted/received with MSB first 1: Data is transmitted/received with LSB first | | | | | | | | | | | | | | | | | | | | |
| 17 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | |

Table 13.17 CSIGNCFG0 Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--|----------|---|--------------------|----------|--------------------------------|---|---|--|---|---|--|---|---|--|---|---|--|
| 16 | CSIGNDAP | Data phase selection bit Select a data phase together with the CSIGNCTL1.CSIGNCKR. See the following table for clock phase and data phase. | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>CSIGNCTL1.CSIGNCKR</th> <th>CSIGNDAP</th> <th>Clock and Data Phase Selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td> </td> </tr> <tr> <td>0</td> <td>1</td> <td> </td> </tr> <tr> <td>1</td> <td>0</td> <td> </td> </tr> <tr> <td>1</td> <td>1</td> <td> </td> </tr> </tbody> </table> | | | CSIGNCTL1.CSIGNCKR | CSIGNDAP | Clock and Data Phase Selection | 0 | 0 | | 0 | 1 | | 1 | 0 | | 1 | 1 | |
| CSIGNCTL1.CSIGNCKR | CSIGNDAP | Clock and Data Phase Selection | | | | | | | | | | | | | | | |
| 0 | 0 | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | | | | | | |
| 15 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.9 CSIGNTX0W — CSIGN Transmission Register 0 for Word Access

This register stores the transmission data. This register also specifies extended data length.

Access: This register can be read/written in 32-bit units.

Address: <CSIGN_base> + 1004_H

Value after reset: 0000 0000_H This register is initialized by a reset from any source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | CSIGNEDL | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | CSIGNTX[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 13.18 CSIGNTX0W Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 | CSIGNEDL | Specifies the extended data length configuration: 0: Normal operation 1: Extended data length enabled The associated data is transmitted as 16-bit data. This bit can only be set if CSIGNCTL1.CSIGNEDLE = 1. |
| 28 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | CSIGNTX[15:0] | Data to be transmitted |

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.10 CSIGNTX0H — CSIGN Transmission Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 in the CSIGNTX0W register.

The setting of 16 high-order bits in CSIGNTX0W is used for data transfer.

Access: This register can be read/written in 16-bit units.

Address: <CSIGN_base> + 1008_H

Value after reset: 0000_H This register is initialized by a reset from any source.

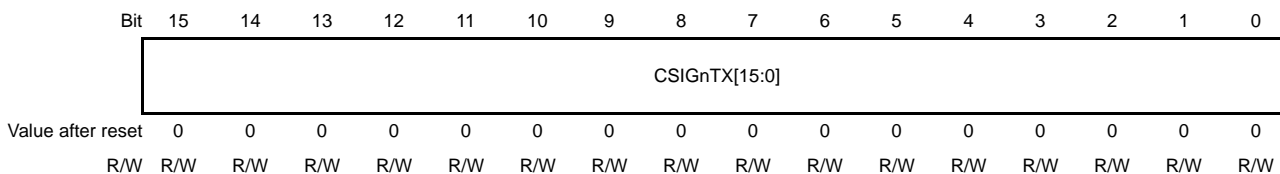


Table 13.19 CSIGNTX0H Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|------------------------|
| 15 to 0 | CSIGNTX[15:0] | Data to be transmitted |

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.11 CSIGNRX0 — CSIGN Reception Register 0

This register stores the received data.

Access: This register can only be read in 16-bit units.

Address: <CSIGN_base> + 100C_H

Value after reset: 0000_H This register is initialized by a reset from any source.

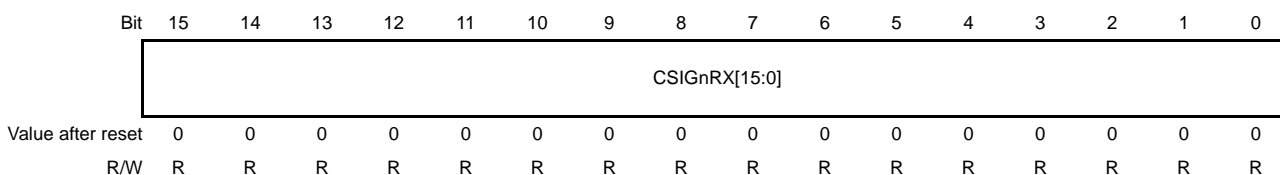


Table 13.20 CSIGNRX0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|---------------|
| 15 to 0 | CSIGNRX [15:0] | Received data |

NOTE

This register stores the received data when an INTCSIG0IR interrupt is generated. Read the received data stored in this register before generation of an INTCSIG0IR interrupt. Otherwise the data is rewritten with the next received data.

CAUTION

When setting this register, refer to **Table 13.21, List of Cautions when Setting Registers.**

13.3.12 List of Caution

Table 13.21 List of Cautions when Setting Registers

| Register Name | Bit Name | Contents |
|------------------------|---|--|
| CSIGNCTL0 | CSIGNPWR | If this bit is cleared during communication, ongoing communication is aborted. After communication is aborted, it is necessary to restart the communication. |
| CSIGNCTL0 | CSIGNTXE CSIGNRXE | Do not modify any of these bits while CSIGNCTL0.CSIGNPWR = 0. (These bits can be modified at the same time as the CSIGNCTL0.CSIGNPWR bit.) Do not modify these bits while CSIGNSTR0.CSIGNTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted. |
| CSIGNCTL0 | CSIGNMBS | When writing, be sure to set to this bit to 1. (The value after reset is "0".) This bit must be modified simultaneously with CSIGNCTL0.CSIGNPWR bit. |
| CSIGNCTL1 | CSIGNCKR | Modification of this bit is only permitted while CSIGNCTL0.CSIGNPWR = 0. |
| CSIGNCTL1 | CSIGNSLIT CSIGNEDLE CSIGNDCS CSIGNHSE | Modification of these bits is only permitted while CSIGNCTL0.CSIGNPWR = 0. |
| CSIGNCTL1 | CSIGNLBM | Modification of this bit is only permitted while CSIGNCTL0.CSIGNPWR = 0. Setting of this bit is prohibited in slave mode. |
| CSIGNCTL1 | CSIGNSIT | Modification of this bit is only permitted while CSIGNCTL0.CSIGNPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated. |
| CSIGNCTL2 | CSIGNPRS[2:0] CSIGNBRS[11:0] | Modification of these bits is only permitted while CSIGNCTL0.CSIGNPWR = 0. Setting of the maximum transfer clock frequency is as follows. <ul style="list-style-type: none"> • Master mode: 8 MHz or less • Slave mode: 6.6 MHz or less |
| CSIGNSTR0 | CSIGNTSF | Writing to this bit is prohibited. Only reading this bit is enabled. |
| CSIGNSTR0 | CSIGNDCE CSIGNPE CSIGNOVE | Writing to these bits is prohibited. Only reading these bits is enabled. These bits are initialized when CSIGNCTL0.CSIGNPWR changes from 0 to 1 or from 1 to 0. |
| CSIGNBCTL0 | CSIGNSCE | To stop the next reception, this bit must be controlled at the following times. <ul style="list-style-type: none"> • When CSIGNSLIT is 0, clear this bit before reception of the last of the data (reading of that data from the CSIGNRX0 register). • When CSIGNSLIT is 1, clear this bit immediately after the second last data reception (reading of that data from the CSIGNRX0 register), up to 1 cycle of the CSIGNTSCK clock before the last CSIGNTIR (reception complete) interrupt. |
| CSIGNCFG0 | CSIGNPS[1:0] CSIGNDLS[3:0] CSIGNDIR CSIGNDAP | Modification of these bits is only permitted while CSIGNCTL0.CSIGNPWR = 0. |
| CSIGNTX0W | CSIGNEDL | This bit is valid only when CSIGNCTL1.CSIGNEDLE = 1. |
| CSIGNTX0W CSIGNTX0H | | Write access to these bits are prohibited when CSIGNCTL0.CSIGNTXE = CSIGNCTL0.CSIGNRXE = 0. |
| CSIGNRX0 | | These bits are initialized when CSIGNPWR in CSIGNCTL0 changes from 0 to 1 or from 1 to 0. It is prohibited to read these bits when CSIGNCTL0.CSIGNTXE = 0 and CSIGNCTL0.CSIGNRXE = 0. |

13.4 Interrupt Sources

CSIG can generate the following interrupts:

- INTCSIG0IC (Communication status interrupt)
- INTCSIG0IR (Reception status interrupt)
- INTCSIG0IRE (Communication error interrupt)

13.4.1 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by a half cycle of the transmission clock CSIGT_{SCK}. This function is not available in slave mode.

The delay is specified by setting bit CSIGN_{CTL1}.CSIGN_{SIT} = 1. (The setting of the CSIGN_{SIT} bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIGN_{CTL1}.CSIGN_{SIT} = 1 (interrupt delay enabled), CSIGN_{CTL1}.CSIGN_{CKR} = 0, CSIGN_{CFG0}.CSIGN_{DAP} = 0 (normal clock and data phase), and CSIGN_{CFG0}.CSIGN_{DLS}[3:0] = 1000_B (data length 8 bits).

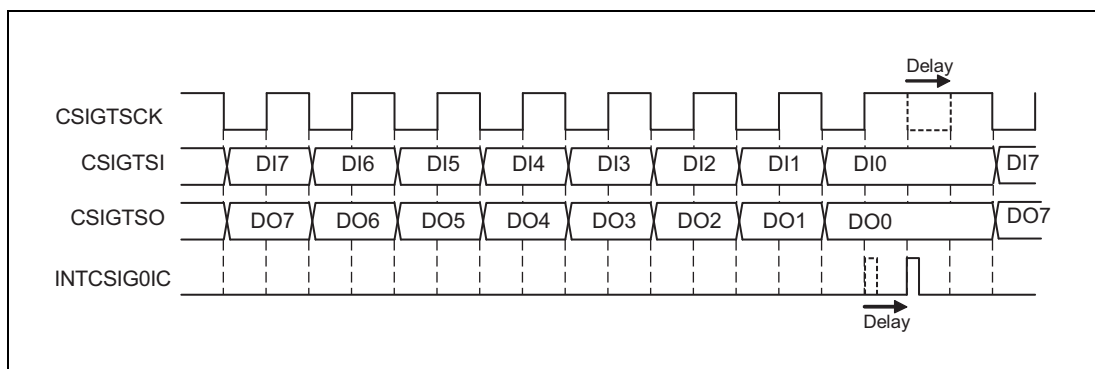


Figure 13.2 Interrupt Delay Function (CSIGN_{CTL1}.CSIGN_{SIT} = 1)

13.4.2 INTCSIG0IC (Communication Status Interrupt)

This interrupt is normally generated after every data transfer. It can be used to trigger a DMA for writing new transmission data to register CSIGNTX0W or CSIGNTX0H.

The following example assumes master mode and a setting of CSIGNCTL1.CSIGNSIT = 0 (no interrupt delay), CSIGNCTL1.CSIGNCKR = 0, CSIGNCFG0.CSIGNDAP = 0 (normal clock and data phase), CSIGNCFG0.CSIGNDLS[3:0] = 1000_B (data length 8 bits), and CSIGNCTL1.CSIGNSLIT = 0 (normal interrupt timing).

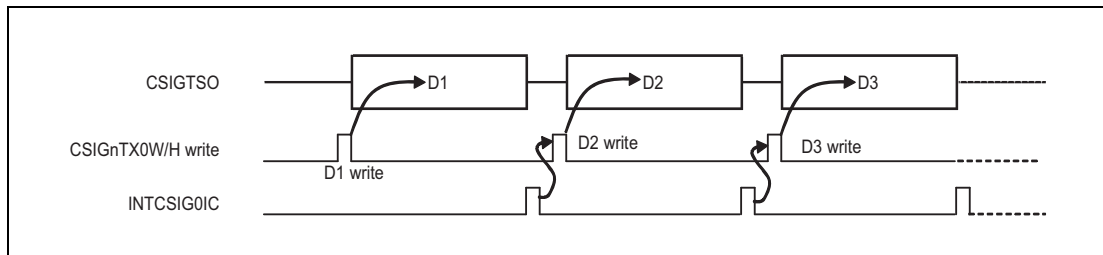


Figure 13.3 Generation of INTCSIG0IC after Communication (CSIGNCTL1.CSIGNSLIT = 0)

However, INTCSIG0IC can also be set up to occur when the CSIGNTX0W/H register is ready to store the next data. This is specified by setting CSIGNCTL1.CSIGNSLIT = 1.

This mode allows more efficient data transfers.

The effect is illustrated in the figure below.

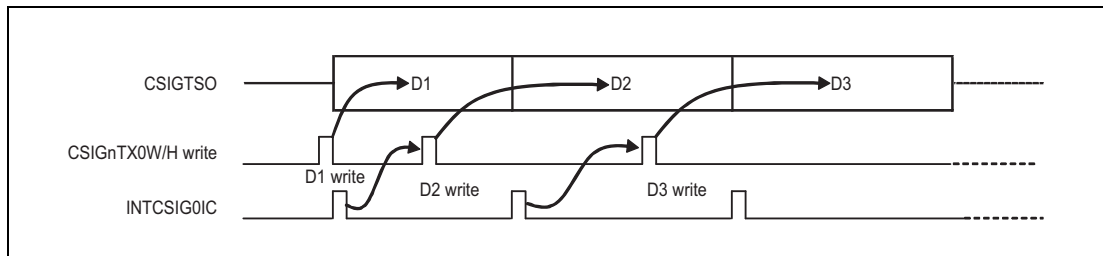


Figure 13.4 Generation of INTCSIG0IC at the Beginning of Communication

13.4.3 INTCSIG0IR (Reception Status Interrupt)

This interrupt is generated in receive-only and transmit/receive mode after data has been received and is available in the reception register. It can be used to trigger a DMA for reading the received data from register CSIGNRX0.

The following example assumes master mode and a setting of CSIGNCTL1.CSIGNSIT = 0 (no interrupt delay), CSIGNCTL1.CSIGNCKR = 0, CSIGNCFG0.CSIGNDAP = 0 (normal clock and data phase), and CSIGNCFG0.CSIGNDLS[3:0] = 1000_B (data length 8 bits).

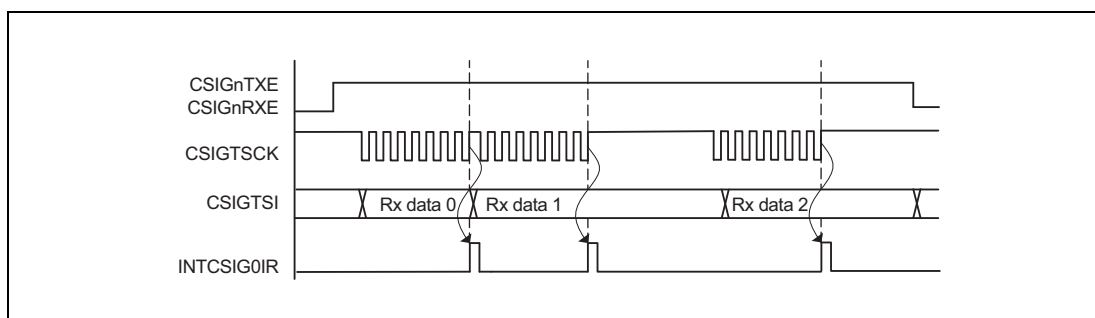


Figure 13.5 Generation of INTCSIG0IR

13.4.4 INTCSIG0IRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

Table 13.22 Data Error Types

| Error type | Communication Status after Error Interrupt | Remarks |
|------------------------------|--|--|
| Parity error | Interrupt is generated and communication continues | — |
| Data consistency check error | Interrupt is generated and communication continues | — |
| Overrun error* ¹ | When CSIGNCTL1.CSIGNHSE = 0 (handshake function disabled) in slave mode, communication continues even after an interrupt is generated. | When CSIGNCTL1.CSIGNHSE = 1 (handshake function enabled) in slave mode, communication stops due to handshake. No interrupt is generated and an overrun error does not occur. |

Note 1. Overrun error does not occur in master mode. In slave mode, communication cannot be stopped.

The type of error that caused the generation of INTCSIG0IRE is indicated in register CSIGNSTR0.

For details about the various error types refer to **Section 13.5.10, Error Detection**.

13.5 Operation

13.5.1 Master/Slave Mode

Whether CSIG operates in master mode or in slave mode depends on the setting of CSIGNCTL2.CSIGNPRS[2:0]. If master mode is selected, the source of the transmission clock must be selected, too.

13.5.1.1 Master Mode

In master mode, the serial communication clock is generated by the internal baud rate generator (BRG) and supplied to the slave via signal CSIGTSCCK.

Master mode is enabled by setting bits CSIGNCTL2.CSIGNPRS[2:0] to values other than 111_B. In master mode, the BRG frequency setting is enabled by setting bits CSIGNCTL2.CSIGNPRS[2:0] and bits CSIGNCTL2.CSIGNBRS[11:0].

The initial level of CSIGTSCCK depends on the CSIGTSCCK clock inversion function bit: it is high level when CSIGNCTL1.CSIGNCKR = 0, and is low level when CSIGNCTL1.CSIGNCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIGNCTL1.CSIGNCKR = 0, CSIGNCFG0.CSIGNDAP = 0, and MSB first:

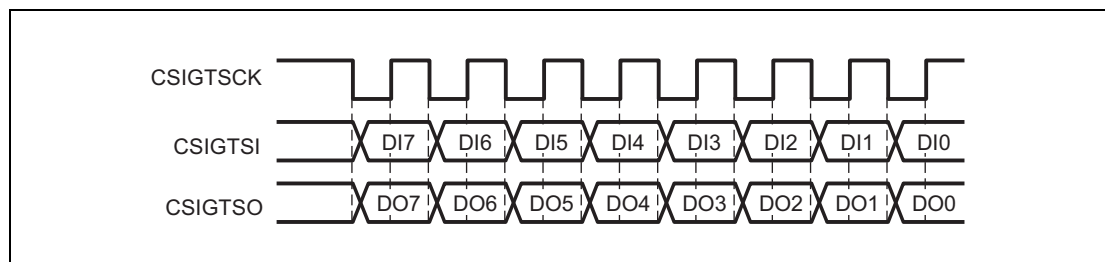


Figure 13.6 Transmit/Receive in Master Mode

13.5.1.2 Slave Mode

In slave mode, another device is the communication master. The external clock is supplied via the signal CSIGTSCK. Transmit/receive operation starts as soon as a clock signal is detected.

Slave mode is selected by setting CSIGnCTL2.CSIGnPRS[2:0] to 111_B.

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting the CSIGnCTL2.CSIGnBRS[11:0] bits to 000_H.

The example below shows the communication in slave mode for 8-bit data, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

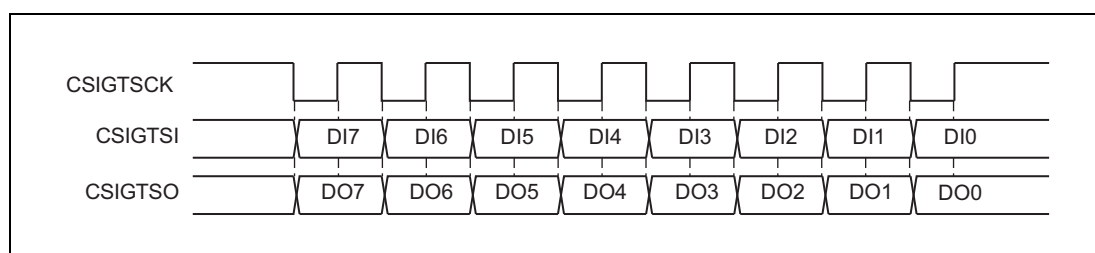


Figure 13.7 Transmit/Receive in Slave Mode

13.5.2 Master/Slave Connections

13.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

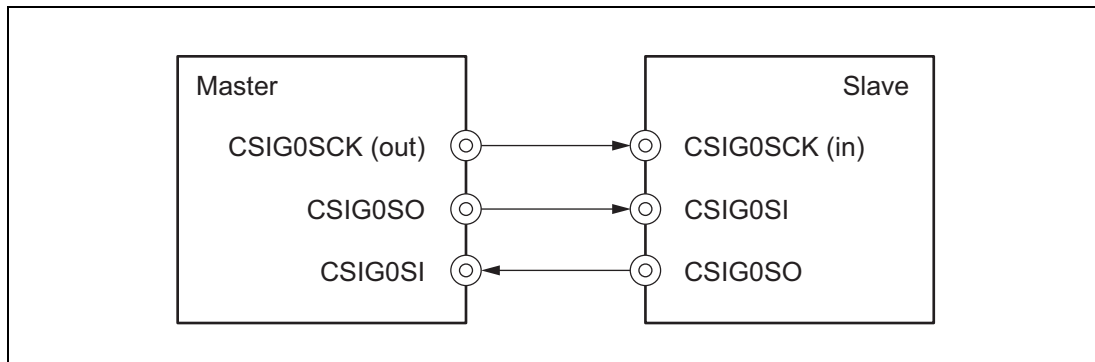


Figure 13.8 Direct Master/Slave Connection

13.5.3 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the CSIGNPRS[2:0] and CSIGNBRS[11:0] bits in the CSIGNCTL2 register.

The following figure shows a block diagram of the BRG.

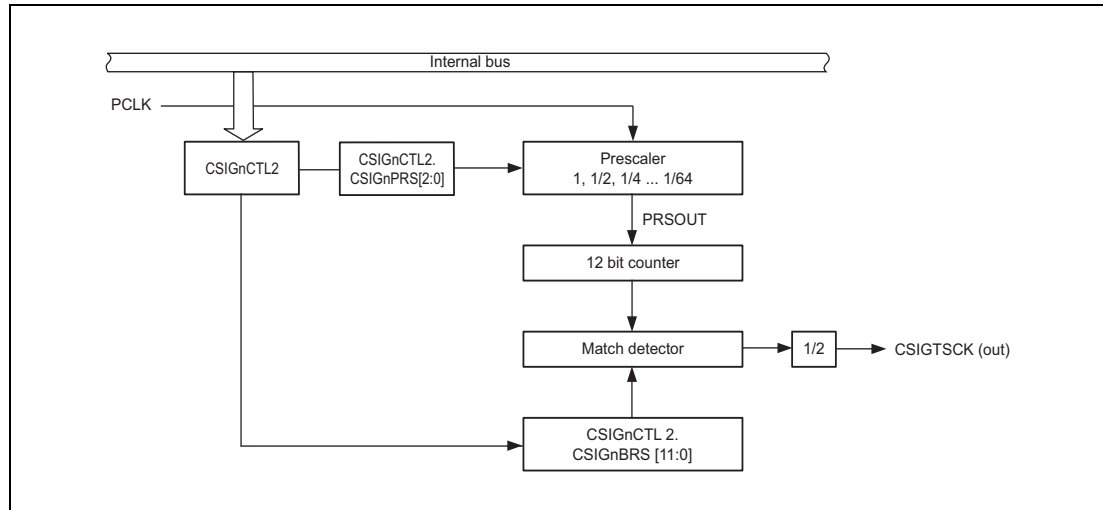


Figure 13.9 BRG Block Diagram

Clearing CSIGNCTL2.CSIGNBRS[11:0] to 000_H disables the BRG.

Calculation of transfer clock frequency

The transfer clock frequency in master mode is calculated by the following formula.

$$\text{Transfer clock frequency (CSIGTSCK)} = \text{PCLK} / (\text{PCLK's division ratio}) = \text{PCLK} / (2^\alpha \times k \times 2)$$

where:

$$\alpha = \text{CSIGNCTL2.CSIGNPRS}[2:0] = 0 \text{ to } 6$$

$$k = \text{CSIGNCTL2.CSIGNBRS}[11:0] = 1 \text{ to } 4095$$

Upper limit and lower limit of transfer clock frequency

Note the following when setting the transfer clock frequency.

- The maximum acceptable transfer clock frequency in master mode is 8 MHz.
- The maximum acceptable transfer clock frequency in slave mode is 6.66 MHz. (It must be confirmed that the transfer clock frequency of the external master is within this range.)
- The minimum transfer clock frequency in both modes is $\text{PCLK} / 524160$.

13.5.4 Data Transfer Modes

13.5.4.1 Transmit-Only Mode

Setting CSIGNCTL0.CSIGNTXE = 1 and CSIGNCTL0.CSIGNRXE = 0 puts the CSIG in transmit-only mode. Transmission starts when transmit data is written in the CSIGNTX0W or CSIGNTX0H register.

CAUTION

In case transmit-only mode has been entered after any reception mode, the data in the CSIGNRX0 buffer becomes undefined after completion of the first transmission. Consequently the reception register CSIGNRX0 has to be read before changing to transmit-only mode.

13.5.4.2 Receive-Only Mode

Setting CSIGNCTL0.CSIGNTXE = 0 and CSIGNCTL0.CSIGNRXE = 1 puts the CSIG in receive-only mode.

In master mode, reception starts when dummy data is read from the CSIGNRX0 register. Subsequent receptions are triggered by reads from the CSIGNRX0 register, as long as CSIGNBCTL0.CSIGNSCE = 1.

Moreover, CSIGNBCTL0.CSIGNSCE has to be set to 0 before reading the last received data from CSIGNRX0.

The recommended procedure is:

1. Set CSIGNBCTL0.CSIGNSCE = 1.
2. Before starting the first receive operation, read CSIGNRX0 (dummy data).
3. Wait for the reception interrupt INTCSIGOIR.
4. Read CSIGNRX0 (received data).
In case more data receptions follow at step 4, continue to read until all data is received.
Before reading the last received data from CSIGNRX0, set CSIGNBCTL0.CSIGNSCE = 0.

In slave mode, reception starts when the communication clock CSIGTSCK from the master is received. In this case, it is not necessary to read data to the CSIGNRX0 register of the slave.

NOTE

In slave mode, any previously received data must be read from the reception register CSIGNRX0 in order to avoid any overwrite situation.

13.5.4.3 Transmit/Receive Mode

Setting CSIGNCTL0.CSIGNTXE = 1 and CSIGNCTL0.CSIGNRXE = 1 puts the CSIG in transmit/receive mode.

Data transfer (transmission and reception) starts when transmit data is written to the CSIGNTX0W or CSIGNTX0H register.

13.5.5 Data Length Selection

13.5.5.1 Data Length Selection Without Extended Length

Transmission data length is selectable from 7 to 16 bits using the CSIGNDLS[3:0] bits in the CSIGNCFG0 register. The examples below show the communication with MSB first (CSIGNCFG0.CSIGNDIR = 0):

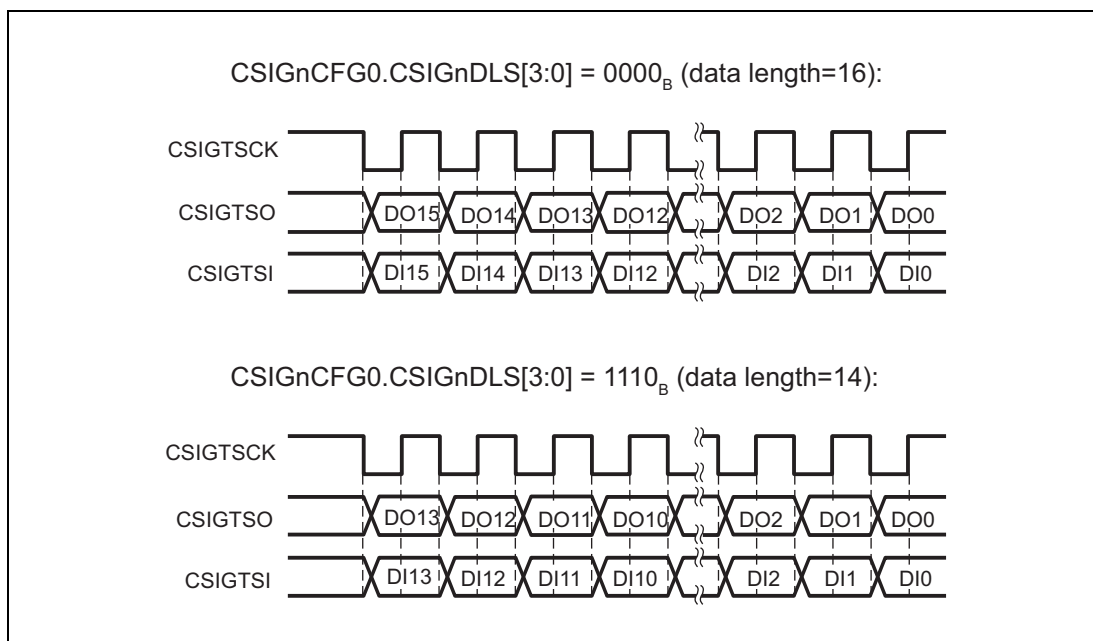


Figure 13.10 Data Length Selection Function

13.5.5.2 Data Length Selection with Extended Data Length

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

The EDL function is enabled by setting bit CSIGNCTL1.CSIGNEDLE to 1.

The EDL function works as follows:

- The data must be divided into 16-bit blocks and remainder. For example, 42-bit data is divided into two 16-bit blocks and 10 bits.
- The remainder bit length is set for CSIGNCFG0.CSIGNDLS[3:0] bits as “data length.”
- CSIGNTX0W.CSIGNEDL must be set to 1 to transmit 16-bit blocks. In this case, the data written to CSIGNTX0W is sent as a 16-bit data length regardless of the CSIGNCFG0.CSIGNDLS[3:0] bits.
- The transfer is complete after the data with the specified data length (the remainder with CSIGNTX0W.CSIGNEDL = 0) has been sent.

Example

Example of transmitting a 40-bit string 123456789A_H.

40 bits are split into 2 × 16 bits plus 8 bits.

- Initialize CSIGNCFG0.CSIGNDLS[3:0] = 8_D.
- To transmit 123456789A_H with MSB first, write the following sequence to CSIGNTX0W.
 - 2000 1234_H (CSIGNTX0W.CSIGNEDL = 1)
 - 2000 5678_H (CSIGNTX0W.CSIGNEDL = 1)
 - 0000 009A_H (CSIGNTX0W.CSIGNEDL = 0)

The following figure illustrates the timing.

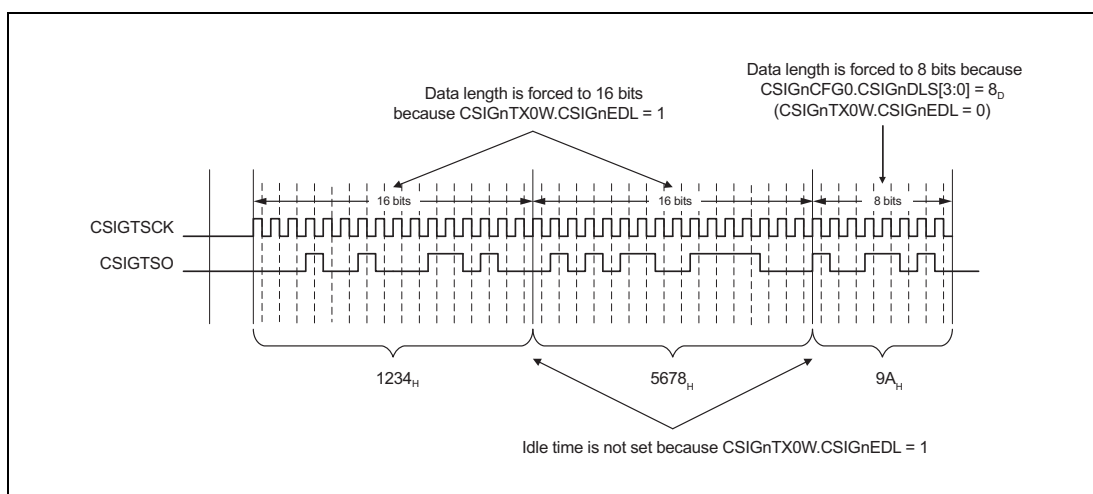


Figure 13.11 EDL Timing Diagram

NOTES

1. A data length of less than 7 bits can be set only when EDL mode is used.
2. It is not possible to transmit two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. The following describes data direction using an example below.
 - Transmit data: 123456_H
 - MSB first:
 - Set CSIGNCFG0.CSIGNDIR to 0.
 - Write CSIGNTX0W = 2000 1234_H. (EDL bit = 1)
 - Write CSIGNTX0W = 0000 0056_H. (EDL bit = 0)
 - LSB first:
 - Set CSIGNCFG0.CSIGNDIR to 1.
 - Write CSIGNTX0W = 2000 3456_H. (EDL bit = 1)
 - Write CSIGNTX0W = 0000 0012_H. (EDL bit = 0)
5. EDL mode is not available in receive-only mode of slave mode. (CSIGNCTL2.CSIGNPRS[2:0] = 111_B, CSIGNCTL0.CSIGNTXE = 0, CSIGNCTL0.CSIGNRXE = 1)

13.5.6 Serial Data Direction Selection Function

The serial data direction is selectable using the CSIGNDIR bit in the CSIGNCFG0 register. The examples below show the communication for 8-bit data (CSIGNCFG0.CSIGNDLS[3:0] = 1000_B):

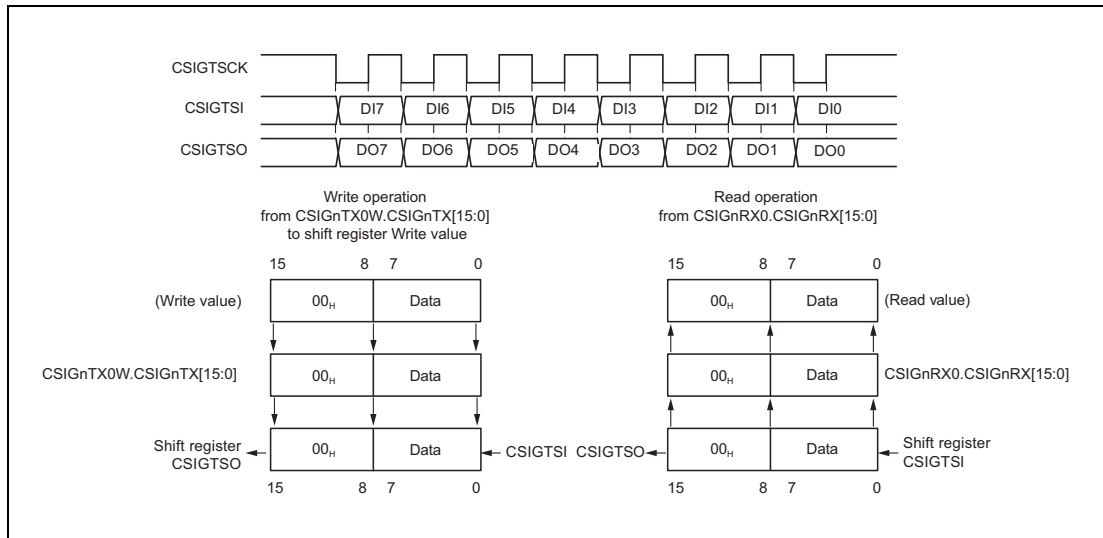


Figure 13.12 Serial Data Direction Select Function — MSB First (CSIGNDIR = 0)

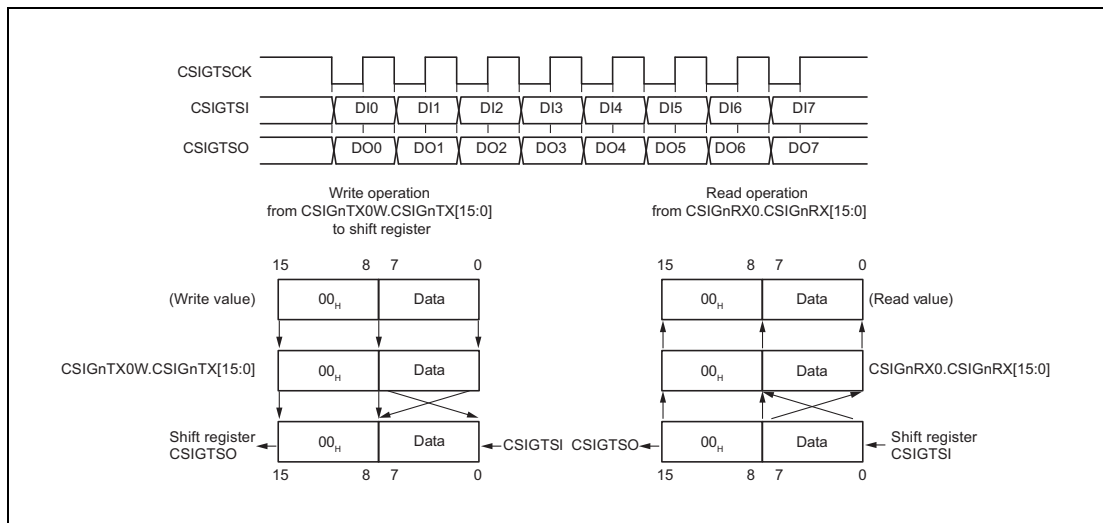


Figure 13.13 Serial Data Direction Select Function — LSB First (CSIGNDIR = 1)

13.5.7 Communication Timing in Slave Mode

The following figure illustrates the communication signals and timings in slave mode.

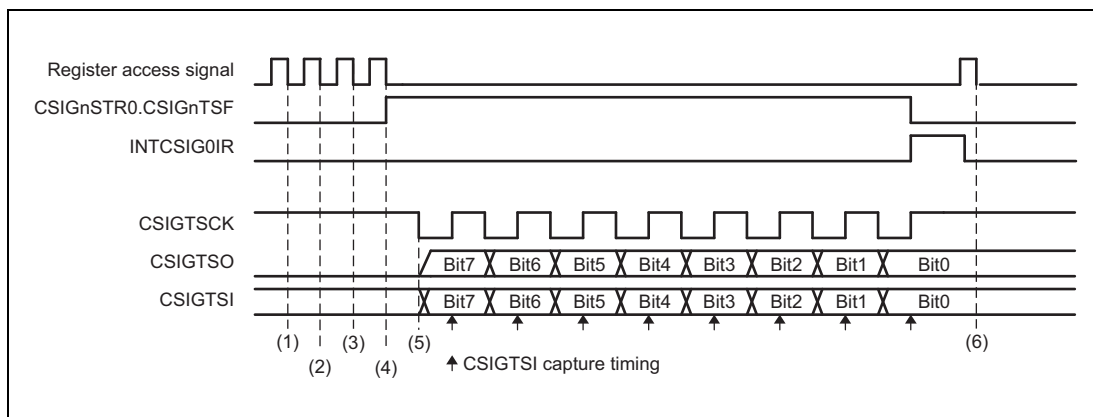


Figure 13.14 Transmit/Receive Communication Timing in Slave Mode

1. CSIG is put into slave mode (Set CSIGnCTL2.CSIGnPRS[2:0] to 111_B)
2. The data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
The data direction is MSB first (CSIGnCFG0.CSIGnDIR = 0)
3. CSIG is set to transmit/receive mode (CSIGnCTL0.CSIGnPWR = 1, CSIGnCTL0.CSIGnTXE = 1, CSIGnCTL0.CSIGnRXE = 1)
4. When transfer data is written to the CSIGnTX0W transmission register or CSIGnTX0H register, the transfer status flag CSIGnSTR0.CSIGnTSF is automatically set.
5. When external clock signal is detected as CSIGTSCK signal, the slave immediately transmits the data to CSIGTSO and, at the same time, captures the data transmitted from CSIGTSI.
6. The CSIGnRX0 register can be read.

13.5.8 Handshake Function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIGNCTL1.CSIGNHSE. For handshake, the CSIGTRYI and CSIGTRYO signals are used.

The timing depends on the setting of the data phase selection bit CSIGNCFG0.CSIGNDAP.

13.5.8.1 Slave Mode

If CSIGNCTL1.CSIGNHSE = 1, a low-level CSIGTRYO signal is output when the slave becomes busy. This happens when previously received data is still in the CSIGNRX0 register, and new data cannot be copied from the shift register to CSIGNRX0 (CSIGNRX0 full condition).

The following examples assume a data length of 8 bits.

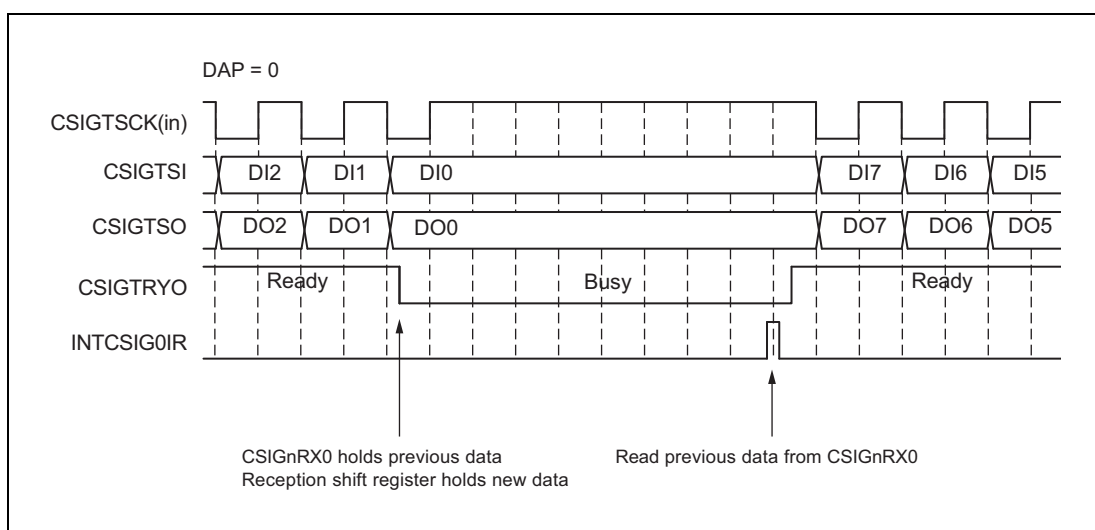


Figure 13.15 Ready/Busy Signal from Slave (CSIGNCFG0.CSIGNDAP = 0)

While the slave is busy, the master has to wait (i.e. suspend the transmission clock). The slave sets CSIGTRYO to high (“ready”) as soon as the reception register CSIGNRX0 has been read.

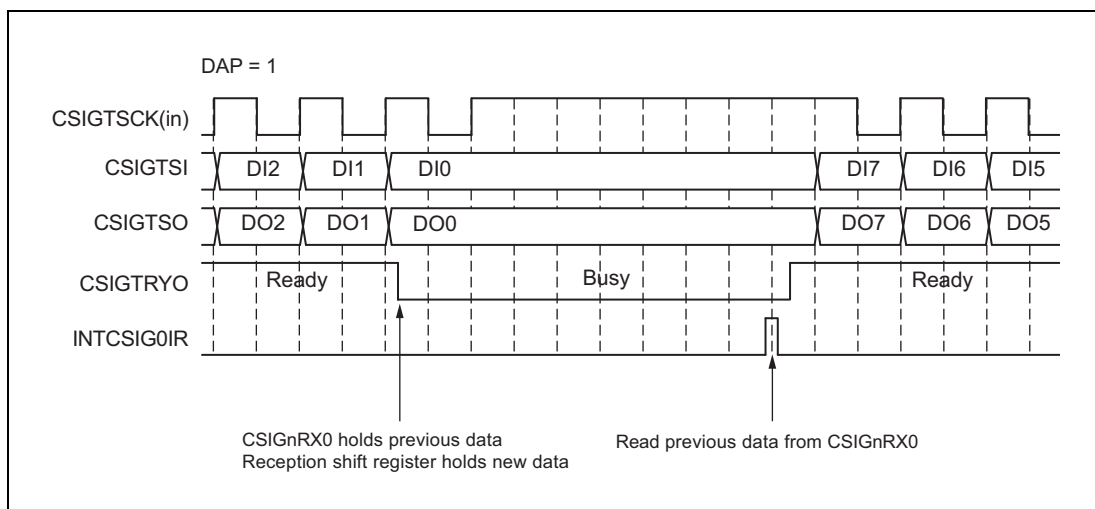


Figure 13.16 Ready/Busy Signal from Slave (CSIGNCFG0.CSIGNDAP = 1)

13.5.8.2 Master Mode

When the master detects low level of the CSIGTRYI while CSIGNCTL1.CSIGNHSE is 1, the subsequent transmission is put on hold, the master goes into wait state and suspends the clock at CSIGTSCK.

The CSIGTRYI level is checked at each half clock cycle of CSIGTSCK.

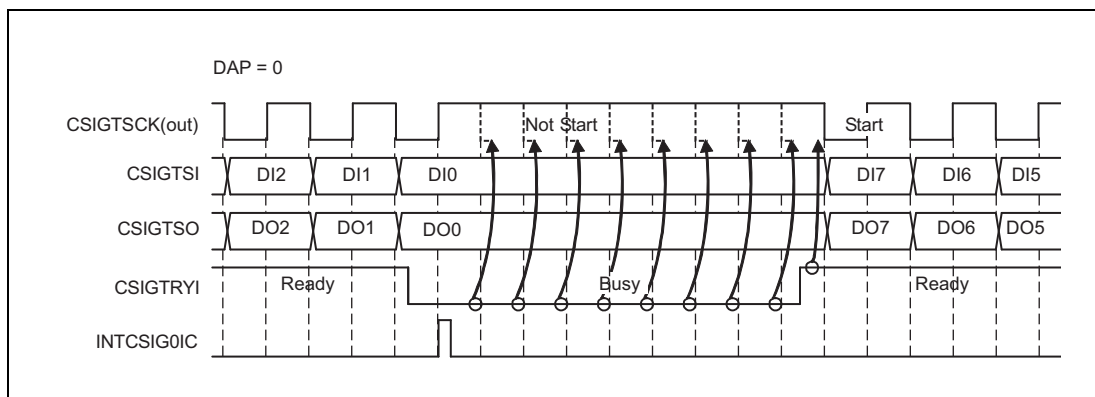


Figure 13.17 Master’s Reaction to CSIGTRYI (CSIGNCFG0.CSIGNDAP = 0)

If the CSIGTRYI low signal from the slave is received while data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIGTRYI becomes high (slave is “ready”).

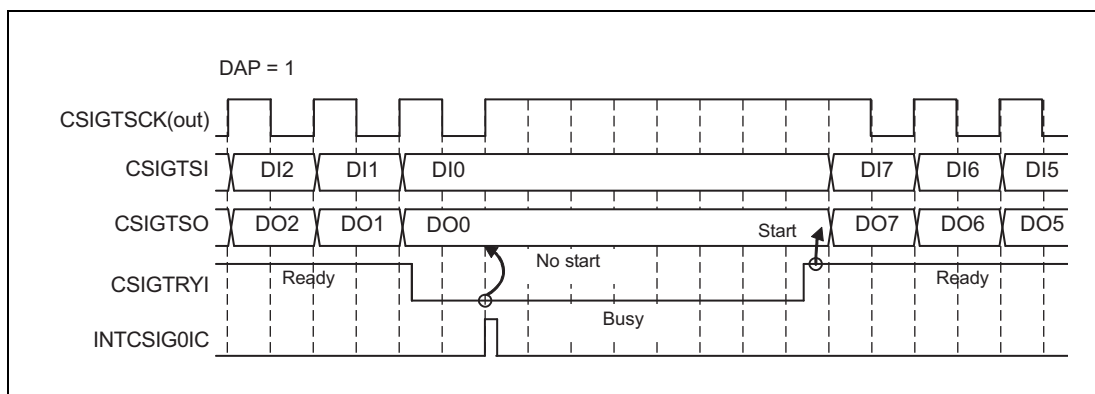


Figure 13.18 Master’s Reaction to CSIGTRYI (CSIGNCFG0.CSIGNDAP = 1)

CAUTION

CSIGTRYI must be pulled down by the slave before the next transfer starts. Even if the signal is pulled down by the slave during the transfer, the transfer will continue until it is finished.

13.5.9 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active ($\text{CSIGNCTL1.CSIGNLBM} = 1$), the transmit and receive signals are internally connected, as shown in the figures below. The signals CSIGTSCCK, CSIGTSO, and CSIGTSI are disconnected from the ports. In addition, the CSIGTSO output level is fixed to low, and CSIGTSCCK is set to the reset level (high). The rest of CSIG works as in normal operation.

In order to test the CSIG, set operating mode to loop-back mode and carry out normal transfer operations.

Table 13.23 Output Level of Pins

| Pin Name | Output Level |
|-----------------|---|
| CSIGTSCCK (out) | High level |
| CSIGTSO | Low level (independent of the last value) |
| Interrupt | Normal function |
| CSIGTRYO | Normal function (low level) |

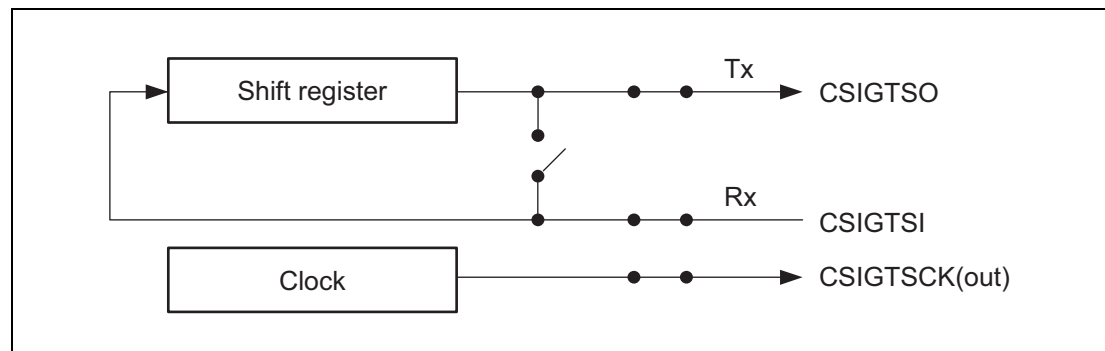


Figure 13.19 Normal Operation

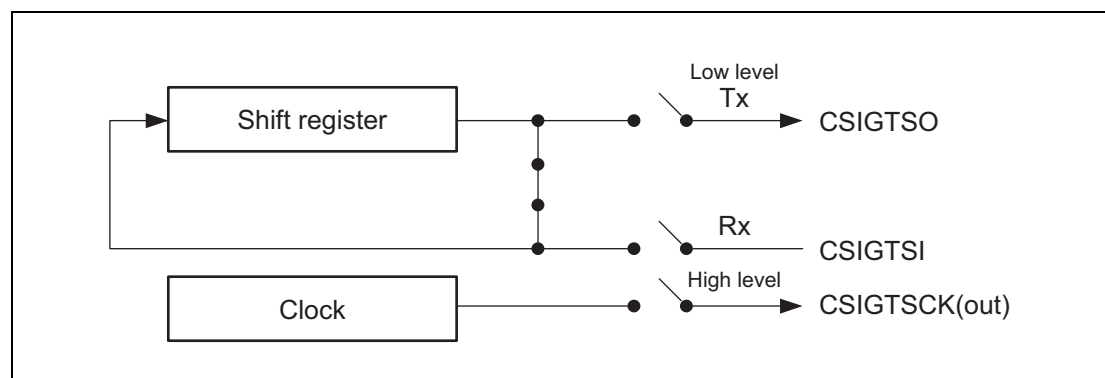


Figure 13.20 Operation in Loop-Back Mode

13.5.10 Error Detection

CSIG can detect three error types:

- Data consistency check error (transmission data)
- Parity error (reception data)
- Overrun error (reception data)

The data consistency check error/parity error check function can be individually enabled or disabled.

If one of these errors is detected, the interrupt INTCSIG0IRE is generated.

13.5.10.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent to the output pin is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by bit CSIGNCTL1.CSIGNDCS. (When executing a data consistency check, always set CSIGTSO to PIPCN.PIPCN_m = 1). It will not be enabled if data transmission is disabled (CSIGNCTL0.CSIGNTXE = 0).

When the data consistency check is enabled, the data transferred from CSIGNTX0W or CSIGNTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIGTSO are captured and the logical interpretation is written to a separate shift register.

After completion of the transmission, the data sent is compared with the original transmission data.

Mismatch is considered as a data consistency check error and:

- Interrupt INTCSIG0IRE is generated.
- Bit CSIGNSTR0.CSIGNDCE is set.

The data consistency check function is illustrated in the following block diagram.

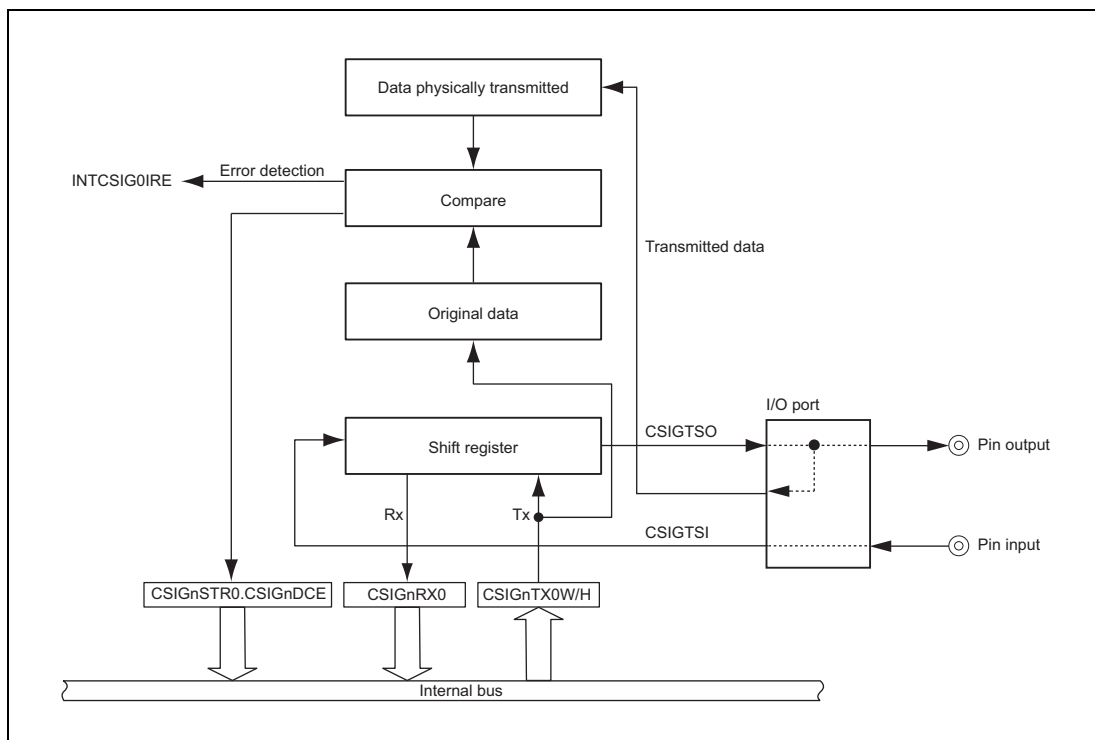


Figure 13.21 Functional Block Diagram of the Data Consistency Check

13.5.10.2 Parity Check

Parity is a common means to detect a single bit error during data transmission. CSIG can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified by CSIGNCFG0.CSIGNPS[1:0].

Parity check is enabled if CSIGNCFG0.CSIGNPS[1] = 1.

The parity bit is checked after reception is complete. If a parity error occurs:

- Interrupt INTCSIG0IRE is generated
- Bit CSIGNSTR0.CSIGNPE is set

The following figure shows an example.

Data length is 8 bits. The data transmitted is 05_H and 35_H. Parity type is odd.

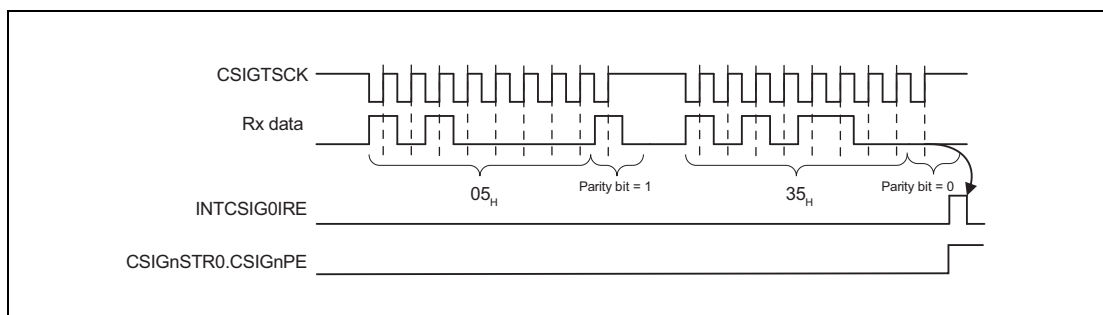


Figure 13.22 Parity Check Example

For the first 8 bits, the parity bit is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

For the second 8 bits, the parity bit is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

When the extended data length (EDL) feature is used, a parity bit is added after the last bit of the data.

13.5.10.3 Overrun Error

This error occurs when previously received data still resides in the reception register CSIGnRX0 because it has not been read and new data is received.

The overrun error is not generated if data reception is disabled (CSIGnCTL0.CSIGnRXE = 0).

If an overrun error occurs:

- Interrupt INTCSIG0IRE is generated.
- Bit CSIGnSTR0.CSIGnOVE is set.
- The data of the CSIGnRX0 register is overwritten and communication continues.

The following figure illustrates the overrun error detection function.

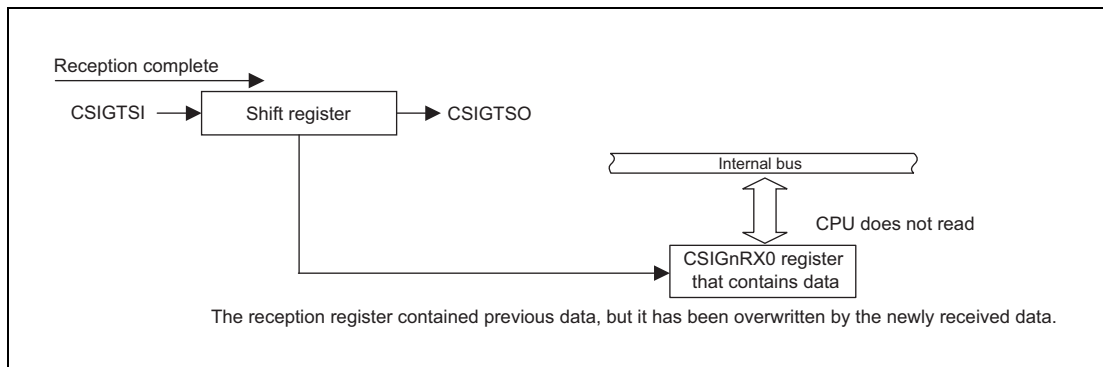


Figure 13.23 Overrun Error Detection

The following figure illustrates an example where:

- Rx data 3 is not read
- Rx data 4 is received and data is overwritten.

Thus an overrun error occurs.

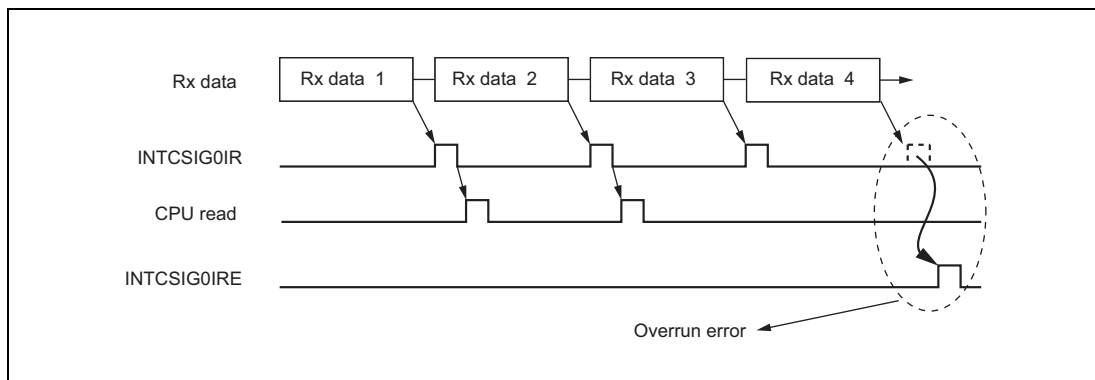


Figure 13.24 Overrun Error Detection - Example

NOTE

An Rx data overrun error can be avoided by using the handshake.

When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

For details see **Section 13.5.8, Handshake Function.**

13.6 Operating Procedures

13.6.1 Transmission/Reception by DMA in Master Mode

This section describes an example of performing the transmission/reception in master mode in combination with a DMA.

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- MSB is transmitted first (CSIGnCFG0.CSIGnDIR = 0)
- INTCSIG0IC interrupt at the end of the transfer (CSIGnCTL1.CSIGnSLIT = 0)
- Normal clock and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- The number of data is 10 (0 to 9)

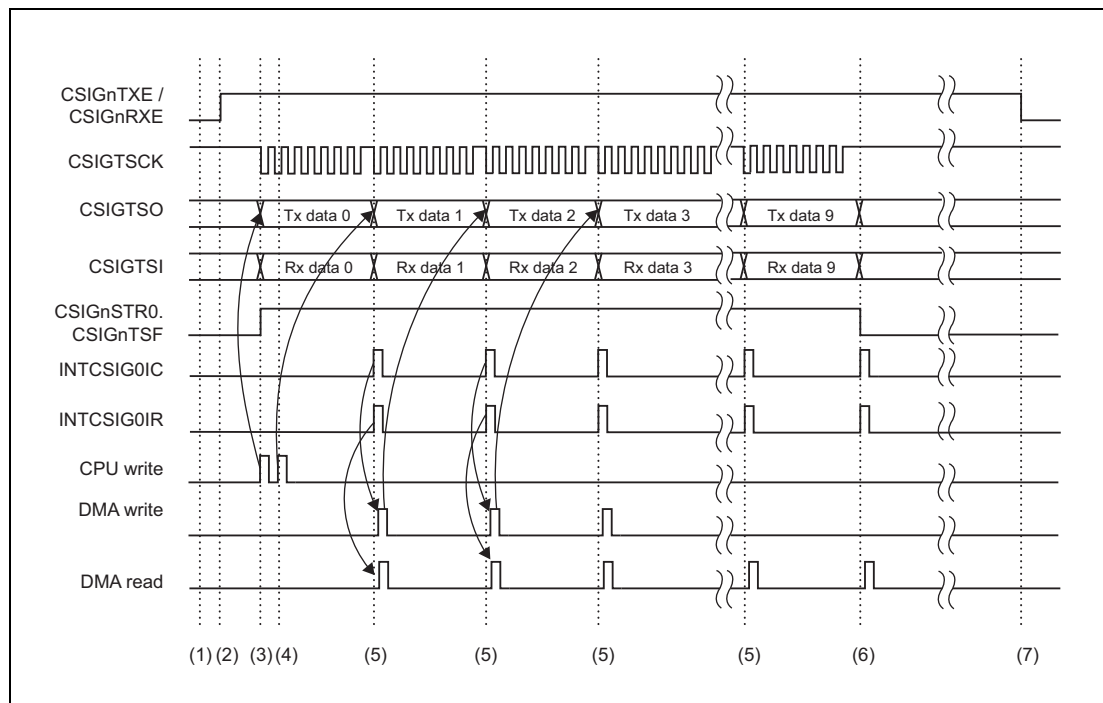


Figure 13.25 Communication in Master Mode

Procedure:

1. Set the communication protocol in the CSIGnCFG0 register. Specify interrupt timing, operating mode, etc. by setting corresponding bits in the CSIGnCTL1 register and the CSIGnCTL2 register.
2. In the CSIGnCTL0 register, set CSIGnPWR = 1 (enable clock), CSIGnTXE = 1 (enable transmission), and CSIGnRXE = 1 (enable reception).
3. Write the first transmit data to be sent to the transmission register CSIGnTX0H. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIGnTX0H. Writing the second data immediately after the first one avoids unnecessary delays between the data.

5. Each time one data block has been transmitted or received, interrupt INTCSIG0IC or INTCSIG0IR is generated. INTCSIG0IC indicates that the next data can be written to CSIGNTX0H.
6. No more write action is required after completion of transmission of data 8. Data 9 (the last data) has been written after transmission of data 7.
However, the reception register CSIGNRX0 must be read after data 8 and data 9 have been received.
7. Finally, to disable the transmit/receive operation, clear CSIGNCTL0.CSIGNTXE and CSIGNCTL0.CSIGNRXE. When no communication is taking place, set CSIGNCTL0.CSIGNPWR to "0" to minimize the power consumption of the CSIGN.

Section 14 Clocked Serial Interface H (CSIH)

The Clocked Serial Interface (CSI) module is a serial data link also well known as SPI, serial peripheral interface. The CSI bus is a synchronous serial data link standard normally operating in full duplex mode via a 3-wire serial I/O. Communication is set up in master/slave mode where the master initiates the data frame and provides the clock. Multiple slave communication is possible with additional individual chip select (slave select) lines. CSIH is implemented 4 units.

14.1 Features of RH850/P1M-E CSIH

14.1.1 Number of Units

This microcontroller has the following number of CSIH units.

Each CSIH unit has one channel interface.

Table 14.1 Number of Units

| Product Name | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 4 | |
| Name | CSIHn (n = 0 to 3) | |

Table 14.2 CSIH Unit Configurations and Channels

| Unit Name (Channel Name) CSIHn | Channels per Unit | RH850/P1M-E 100 pins (4 ch) | RH850/P1M-E 144 pins (4 ch) |
|--------------------------------------|----------------------|-----------------------------------|-----------------------------------|
| CSIH0 | 1 | | √ |
| CSIH1 | 1 | | √ |
| CSIH2 | 1 | | √ |
| CSIH3 | 1 | | √ |

Note: The channel names are same as those of the corresponding units.

The following table lists the values indicated by the index of each product.

Table 14.3 Index

| Index | Description |
|-------|---|
| n | Throughout this section, the individual CSIH units are identified by the index "n" (n = 0 to 3): for example, CSIHnCTL0 is the CSIHn control register0. |
| x | CSIHn has up to 8 chip select signals. Throughout this section, the individual chip select signals are identified by the index "x": that is, CSx denotes a specified chip select signal. For the index "x", refer to Table 14.4, Number of Chip Select Signals . |
| y | A variable used for explanation is identified by the index "y" (y=0 to 3): for example, CSIHnBRSy the baud rate setting register of CSIHn. |

The numbers of chip select signals for each of the CSIH units are listed in the following table.

Table 14.4 Number of Chip Select Signals

| Unit Name | Chip Select Index | |
|-----------|-------------------|------------------|
| | 100 pins | 144 pins |
| CSIH0 | CSx (x = 0 to 7) | CSx (x = 0 to 7) |
| CSIH1 | CSx (x = 0 to 3) | CSx (x = 0 to 7) |
| CSIH2 | CSx (x = 0 to 7) | CSx (x = 0 to 7) |
| CSIH3 | CSx (x = 0 to 7) | CSx (x = 0 to 7) |

14.1.2 Register Base Address

CSIH base addresses are listed in the following table.

CSIH register addresses are given as offsets from the base addresses throughout the section.

Table 14.5 Register Base Addresses

| Base Address Name | Base Address |
|-------------------|------------------------|
| <CSIH0_base> | FFD8 0000 _H |
| <CSIH1_base> | FFD8 2000 _H |
| <CSIH2_base> | FFD8 4000 _H |
| <CSIH3_base> | FFD8 6000 _H |

14.1.3 Clock Supply

Clock supply by and to CSIH is listed in the following table.

Table 14.6 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|-------------------------------------|
| CSIHn | PCLK | High-speed peripheral clock CLK_HSB |

14.1.4 Interrupt Requests

CSIH interrupt requests are listed in the following table.

Table 14.7 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|---|------------------|------------------------|
| CSIH0 | | | |
| INTCSIH0IRE | CSIH0 communication error interrupt | 82 | — |
| INTCSIH0IR0S | CSIH0 receive status/ CS0 receive status interrupt | 83 | 65 |
| INTCSIH0IC0S | CSIH0 communication status/ CS0 communication status interrupt | 84 | 66 |
| INTCSIH0IR1 | CSIH0 CS1 receive status interrupt | 85 | 67 |
| INTCSIH0IC1 | CSIH0 CS1 communication status interrupt | 86 | 68 |
| INTCSIH0IR2 | CSIH0 CS2 receive status interrupt | 87 | 69 |
| INTCSIH0IC2 | CSIH0 CS2 communication status interrupt | 88 | 70 |
| INTCSIH0IJC | CSIH0 job completion interrupt | 89 | 71 |
| CSIH1 | | | |
| INTCSIH1IRE | CSIH1 communication error interrupt | 90 | — |
| INTCSIH1IR0S | CSIH1 receive status/CS0 receive status interrupt | 91 | 72 |
| INTCSIH1IC0S | CSIH1 communication status/CS0 communication status interrupt | 92 | 73 |
| INTCSIH1IR1 | CSIH1 CS1 receive status interrupt | 93 | 74 |
| INTCSIH1IC1 | CSIH1 CS1 communication status interrupt | 94 | 75 |
| INTCSIH1IR2 | CSIH1 CS2 receive status interrupt | 95 | 76 |
| INTCSIH1IC2 | CSIH1 CS2 communication status interrupt | 96 | 77 |
| INTCSIH1IJC | CSIH1 job completion interrupt | 97 | 78 |
| CSIH2 | | | |
| INTCSIH2IRE | CSIH2 communication error interrupt | 98 | — |
| INTCSIH2IR | CSIH2 receive status interrupt | 99 | 79 |
| INTCSIH2IC | CSIH2 communication status interrupt | 100 | 80 |
| INTCSIH2IJC | CSIH2 job completion interrupt | 101 | 81 |
| CSIH3 | | | |
| INTCSIH3IRE | CSIH3 communication error interrupt | 102 | — |
| INTCSIH3IR | CSIH3 receive status interrupt | 103 | 82 |
| INTCSIH3IC | CSIH3 communication status interrupt | 104 | 83 |
| INTCSIH3IJC | CSIH3 job completion interrupt | 105 | 84 |

14.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

14.1.6 External Input/Output Signals

External input/output signals of CSIH are listed below.

Table 14.8 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal |
|------------------------------|-----|----------------------------|------------------------------|
| CSIH0 | | | |
| CSIHTSCK | I | Serial clock signal input | CSIH0SCI |
| | O | Serial clock signal output | CSIH0SCO |
| CSIHTSI | I | Serial data input signal | CSIH0SI |
| $\overline{\text{CSIHTSSI}}$ | I | Slave select input signal | $\overline{\text{CSIH0SSI}}$ |
| CSIHTRYI | I | Ready/busy input signal | CSIH0RYI |
| CSIHTSO | O | Serial data output signal | CSIH0SO |
| CSIHTRYO | O | Ready/busy output signal | CSIH0RYO |
| CSIHTCSS[7:0]* ¹ | O | Chip select signal | CSIH0CSS[7:0]* ¹ |
| CSIH1 | | | |
| CSIHTSCK | I | Serial clock signal input | CSIH1SCI |
| | O | Serial clock signal output | CSIH1SCO |
| CSIHTSI | I | Serial data input signal | CSIH1SI |
| $\overline{\text{CSIHTSSI}}$ | I | Slave select input signal | $\overline{\text{CSIH1SSI}}$ |
| CSIHTRYI | I | Ready/busy input signal | CSIH1RYI |
| CSIHTSO | O | Serial data output signal | CSIH1SO |
| CSIHTRYO | O | Ready/busy output signal | CSIH1RYO |
| CSIHTCSS[7:0]* ¹ | O | Chip select signal | CSIH1CSS[7:0]* ¹ |
| CSIH2 | | | |
| CSIHTSCK | I | Serial clock signal input | CSIH2SCI |
| | O | Serial clock signal output | CSIH2SCO |
| CSIHTSI | I | Serial data input signal | CSIH2SI |
| $\overline{\text{CSIHTSSI}}$ | I | Slave select input signal | $\overline{\text{CSIH2SSI}}$ |
| CSIHTRYI | I | Ready/busy input signal | CSIH2RYI |
| CSIHTSO | O | Serial data output signal | CSIH2SO |
| CSIHTRYO | O | Ready/busy output signal | CSIH2RYO |
| CSIHTCSS[7:0]* ¹ | O | Chip select signal | CSIH2CSS[7:0]* ¹ |
| CSIH3 | | | |
| CSIHTSCK | I | Serial clock signal input | CSIH3SCI |
| | O | Serial clock signal output | CSIH3SCO |
| CSIHTSI | I | Serial data input signal | CSIH3SI |
| $\overline{\text{CSIHTSSI}}$ | I | Slave select input signal | $\overline{\text{CSIH3SSI}}$ |
| CSIHTRYI | I | Ready/busy input signal | CSIH3RYI |
| CSIHTSO | O | Serial data output signal | CSIH3SO |
| CSIHTRYO | O | Ready/busy output signal | CSIH3RYO |
| CSIHTCSS[7:0]* ¹ | O | Chip select signal | CSIH3CSS[7:0]* ¹ |

Note 1. For the number of chip select signals, refer to **Table 14.4, Number of Chip Select Signals**.

14.1.7 Data Consistency Check

The following table lists the port pins on which CSIHnSO pin functions are multiplexed and whether or not the CSIHnSO pin functions support data consistency checking. Refer to **Section 14.5.12, Error Detection** for details on data consistency checking.

Table 14.9 Data Consistency Checking and Port Pins

| Unit Signal Name | Port Pin Name | Alternative Function | Data Consistency Checking |
|------------------|---------------|----------------------|---------------------------|
| CSIH0 | | | |
| CSIHTSO | P2_5 | ALT-OUT6 | Supported |
| | P2_12*1 | ALT-OUT6 | Supported |
| | P3_0*1 | ALT-OUT5 | Supported |
| | P3_7 | ALT-OUT1 | Supported |
| CSIH1 | | | |
| CSIHTSO | P2_8 | ALT-OUT3 | Supported |
| | P4_0 | ALT-OUT1 | Supported |
| CSIH2 | | | |
| CSIHTSO | P1_3 | ALT-OUT4 | Supported |
| | P2_1 | ALT-OUT2 | Supported |
| | P4_5 | ALT-OUT1 | Supported |
| | P5_14 | ALT-OUT5 | Not supported |
| CSIH3 | | | |
| CSIHTSO | P1_3 | ALT-OUT6 | Supported |
| | P2_5 | ALT-OUT4 | Supported |

Note 1. Available for 144-pin devices.

14.1.8 Combinations of Pins and Ports

Combinations of CSIH pins and ports are listed in the following table.

Table 14.10 Combinations of pins and ports (1/2)

| Function | Pin Name | Port Name | | | |
|----------|-----------------------|--|---------|--------------------|---------------------|
| | | Group 1 | Group 2 | Group 3 | Group 4 |
| CSIH0 | CSIH0CSS0 | P3_12 | | | |
| | CSIH0CSS1 | P3_13 | | | |
| | CSIH0CSS2 | P2_4 / P2_14* ¹ | | | |
| | CSIH0CSS3 | P2_5 / P2_13* ¹ | | | |
| | CSIH0CSS4 | P2_6 / P2_12* ¹ | | | |
| | CSIH0CSS5 | P2_2 / P2_7 / P2_11* ¹ | | | |
| | CSIH0CSS6 | P2_8 / P2_10* ¹ | | | |
| | CSIH0CSS7 | P2_9 / P3_2* ¹ | | | |
| | CSIH0SSI | P3_8 | | | |
| | CSIH0RYI | P2_10* ¹ / P2_15* ¹ / P3_3 / P3_9 | | | |
| | CSIH0RYO | P2_11* ¹ | | | |
| | CSIH0SCI/ CSIH0SCO | P2_6 | P3_8 | P3_1* ¹ | P3_1* ¹ |
| | CSIH0SI | P2_4 | P3_6 | P3_2* ¹ | P3_6 |
| | CSIH0SO | P2_5 | P3_7 | P3_0* ¹ | P2_12* ¹ |
| CSIH1 | CSIH1CSS0 | P4_3 | | — | — |
| | CSIH1CSS1 | P4_4 | | — | — |
| | CSIH1CSS2 | P4_5 | | — | — |
| | CSIH1CSS3 | P4_6 | | — | — |
| | CSIH1CSS4 | P4_7* ¹ | | — | — |
| | CSIH1CSS5 | P4_8* ¹ | | — | — |
| | CSIH1CSS6 | P4_9* ¹ | | — | — |
| | CSIH1CSS7 | P4_10* ¹ | | — | — |
| | CSIH1SSI | P4_8* ¹ | | — | — |
| | CSIH1RYI | P2_11* ¹ / P3_13 / P4_9* ¹ | | — | — |
| | CSIH1RYO | P2_13* ¹ / P3_12 / P4_2 / P4_10* ¹ | | — | — |
| | CSIH1SCI/ CSIH1SCO | P2_9 | P4_1 | — | — |
| | CSIH1SI | P2_7 | P3_14 | — | — |
| | CSIH1SO | P2_8 | P4_0 | — | — |

Table 14.10 Combinations of pins and ports (2/2)

| Function | Pin Name | Port Name | | | |
|----------|-----------------------|--|---------|---------|---------------------|
| | | Group 1 | Group 2 | Group 3 | Group 4 |
| CSIH2 | CSIH2CSS0 | P1_1 / P2_3 / P4_7 ^{*1} /P5_2 ^{*1} | | | |
| | CSIH2CSS1 | P3_3 / P4_8 ^{*1} / P5_0 | | | |
| | CSIH2CSS2 | P3_4 / P4_9 ^{*1} / P5_1 | | | |
| | CSIH2CSS3 | P3_5 / P4_10 ^{*1} / P5_4 | | | |
| | CSIH2CSS4 | P3_6 / P4_11 ^{*1} / P5_5 | | | |
| | CSIH2CSS5 | P3_7 / P4_12 ^{*1} / P5_6 | | | |
| | CSIH2CSS6 | P3_8 / P4_14 ^{*1} / P5_7 ^{*1} | | | |
| | CSIH2CSS7 | P3_9 / P4_4 / P5_8 ^{*1} | | | |
| | CSIH2SSI | P4_13 ^{*1} /P5_11 ^{*1} | | | |
| | CSIH2RYI | P2_3 / P3_7 / P4_3/P5_12 ^{*1} | | | |
| | CSIH2RYO | P2_4 / P4_7 ^{*1} /P5_13 ^{*1} | | | |
| | CSIH2SCI/ CSIH2SCO | P1_4 | P2_2 | P4_6 | P5_15 ^{*1} |
| | CSIH2SI | P1_2 | P2_0 | P4_4 | P5_9 |
| | CSIH2SO | P1_3 | P2_1 | P4_5 | P5_14 |
| CSIH3 | CSIH3CSS0 | P2_8 | | — | — |
| | CSIH3CSS1 | P2_9 | | — | — |
| | CSIH3CSS2 | P3_5 | | — | — |
| | CSIH3CSS3 | P3_4 | | — | — |
| | CSIH3CSS4 | P3_3 | | — | — |
| | CSIH3CSS5 | P3_9 | | — | — |
| | CSIH3CSS6 | P2_0 | | — | — |
| | CSIH3CSS7 | P2_1 | | — | — |
| | CSIH3SSI | P3_3 | | — | — |
| | CSIH3RYI | P1_1 / P2_8 / P3_6 | | — | — |
| | CSIH3RYO | P1_0 ^{*1} / P2_8 | | — | — |
| | CSIH3SCI/ CSIH3SCO | P1_4 | P2_7 | — | — |
| | CSIH3SI | P1_2 | P2_6 | — | — |
| | CSIH3SO | P1_3 | P2_5 | — | — |

Note 1. Available in devices with 144-pin.

14.2 Overview

14.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode and slave mode selectable
- Supports multiple slaves configuration and RCB (Recessive Configuration for Broadcasting) thanks to eight configurable chip select output signals
- Slave select input signal ($\overline{\text{CSIHTSSI}}$) is available.
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, and in slave mode, it is determined by input clock.
- Maximum transfer clock frequency:
 - in master mode: 10 MHz
 - in slave mode: 6.66 MHz
- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable from 2 to 16 bits in 1-bit units
- Provided with on-chip EDL (Extended Data Length) function for transferring data with more than 16 bits.
- Three selectable transfer modes:
 - transmit-only mode
 - receive-only mode
 - transmit/receive mode
- Built-in handshake function
- Provided with on-chip error detection function (data consistency check, parity, timeout, overflow, and overrun).
- Supporting job concept
- 128 words I/O buffer memory
- Direct access mode and memory mode (FIFO, dual buffer, transmit-only buffer) is selectable.
- Four different interrupt request signals (INTCSIHnIC, INTCSIHnIR, INTCSIHnIRE, INTCSIHnJC)
- Provided with on-chip LBM (Loop Back Mode) function for self-test.
- CPU-controlled high-priority communication function
- Enforced chip select idle setting
- Provided with RCB (Recessive Configuration for Broadcasting) bit for broadcasting.

14.2.2 Functional Overview Description

The CSIH uses three signals for communication:

- Transmission clock CSIH_TSCK (output in master mode, input in slave mode)
- Data output signal CSIH_TSO
- Data input signal CSIH_TSI

Additional signals are available for external control and monitoring:

- $\overline{\text{CSIH}}\text{TSSI}$: Slave select input signal
- CSIH_TRYO: Ready/busy output signal (handshake signal)
- CSIH_TRYI: Ready/busy input signal (handshake signal)
- CSIH_TCSS[7:0]: Chip select signals

Data transmission is bit-wise and serial, and synchronous to the transmission clock.

The following table shows the most important registers for setting up the CSIH.

Table 14.11 Main Registers of CSIH

| Register | Function |
|------------|--|
| CSIHnCTL0 | Enables/disables serial clock, and permits/ prohibits data transmission and data reception. Defines end-of-job behavior and enables/disables buffering (bypass of the buffer). |
| CSIHnCTL1 | Controls options such as interrupt timing, extended data length, job feature, data consistency check, loop-back mode, handshake, etc. |
| CSIHnCTL2 | Selects master mode or slave mode, and selects transfer clock frequency for the on-chip baud rate generator (BRG) in master mode. |
| CSIHnBRSy | Sets transfer clock frequency for each chip select signal. |
| CSIHnMCTL0 | Selects memory mode and specifies the time-out value. |
| CSIHnMCTL1 | Controls the memory in FIFO mode. |
| CSIHnMCTL2 | Controls the memory in dual buffer mode. |
| CSIHnCFGx | Registers to configure the communication protocol for each chip select signal. |

14.2.3 Block Diagram

The following block diagram shows the main components of the CSIH.

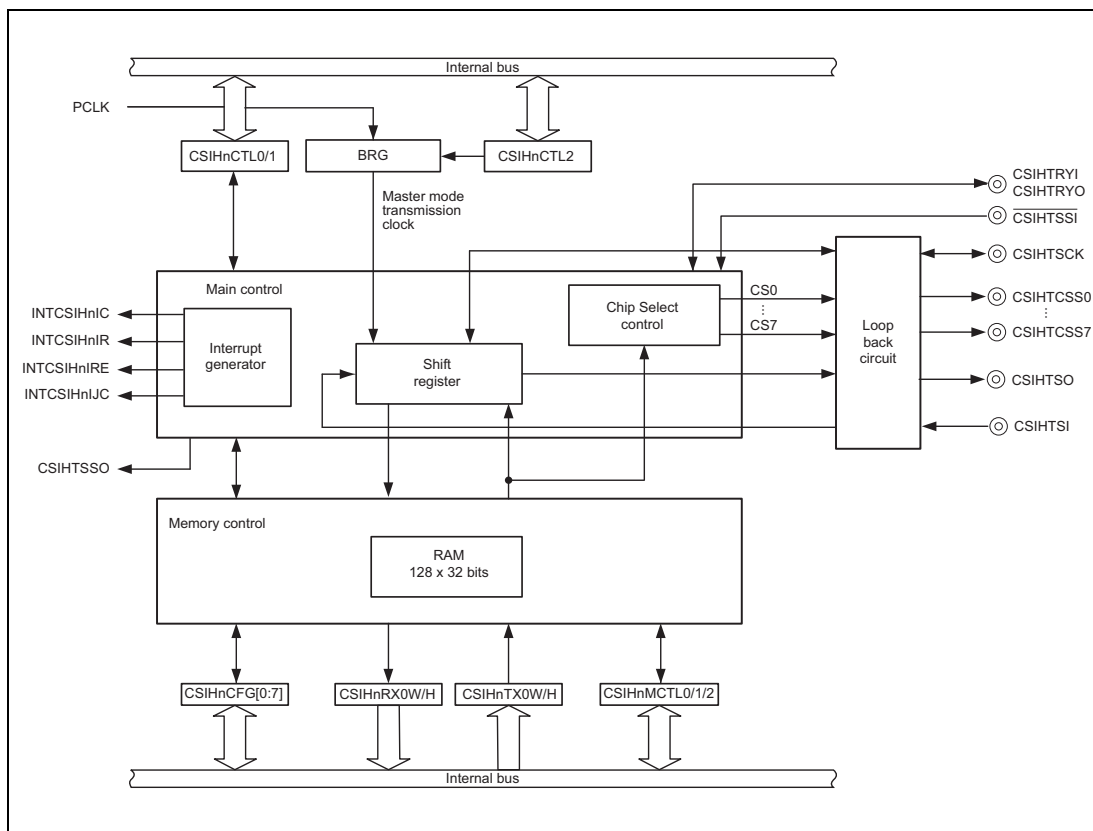


Figure 14.1 CSIH Block Diagram

In master mode, the transmission clock CSIHnTSCk is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

The built-in memory can be configured as FIFO, dual buffer (separate transmit and receive buffers), or transmit-only buffer. It can also be bypassed for data transmission and reception without buffering.

The loop back circuit disconnects the CSIH completely from the ports and supports internal self-test.

NOTE

This chapter describes the following modes:

- The “operating mode” is either master mode or and slave mode. In this context, only a master can control and communicate with several slaves (for details, see **Section 14.5.1, Operating Modes (Master/Slave)**).
- The “job mode” is related to the AUTOSAR job concept (for details, see **Section 14.5.3.3, Job Concept**).
- The “memory mode” accords with the various configurations of the associated buffer memory (for details, see **Section 14.5.6, CSIH Buffer Memory**).
- The “data transfer mode” specifies the type of communication – transmit-only, receive only, or transmit/receive (for details, see **Section 14.5.7, Data Transfer Modes**).

14.3 Registers

14.3.1 List of Registers

CSIH registers are listed in the following table.

The base addresses <CSIH_base> of the CSIHn is defined in **Section 14.1.2, Register Base Address**.

Table 14.12 List of Registers

| Module | Register Name | Symbol | Address |
|--------|---|------------|----------------------------------|
| CSIHn | CSIHn control register 0 | CSIHnCTL0 | <CSIHn_base> + 0000 _H |
| CSIHn | CSIHn control register 1 | CSIHnCTL1 | <CSIHn_base> + 0010 _H |
| CSIHn | CSIHn control register 2 | CSIHnCTL2 | <CSIHn_base> + 0014 _H |
| CSIHn | CSIHn status register 0 | CSIHnSTR0 | <CSIHn_base> + 0004 _H |
| CSIHn | CSIHn status clear register 0 | CSIHnSTCR0 | <CSIHn_base> + 0008 _H |
| CSIHn | CSIHn memory control register 0 | CSIHnMCTL0 | <CSIHn_base> + 1040 _H |
| CSIHn | CSIHn memory control register 1 | CSIHnMCTL1 | <CSIHn_base> + 1000 _H |
| CSIHn | CSIHn memory control register 2 | CSIHnMCTL2 | <CSIHn_base> + 1004 _H |
| CSIHn | CSIHn memory read/write pointer register 0 | CSIHnMRWP0 | <CSIHn_base> + 1018 _H |
| CSIHn | CSIHn configuration register 0 | CSIHnCFG0 | <CSIHn_base> + 1044 _H |
| CSIHn | CSIHn configuration register 1 | CSIHnCFG1 | <CSIHn_base> + 1048 _H |
| CSIHn | CSIHn configuration register 2 | CSIHnCFG2 | <CSIHn_base> + 104C _H |
| CSIHn | CSIHn configuration register 3 | CSIHnCFG3 | <CSIHn_base> + 1050 _H |
| CSIHn | CSIHn configuration register 4 | CSIHnCFG4 | <CSIHn_base> + 1054 _H |
| CSIHn | CSIHn configuration register 5 | CSIHnCFG5 | <CSIHn_base> + 1058 _H |
| CSIHn | CSIHn configuration register 6 | CSIHnCFG6 | <CSIHn_base> + 105C _H |
| CSIHn | CSIHn configuration register 7 | CSIHnCFG7 | <CSIHn_base> + 1060 _H |
| CSIHn | CSIHn transmit data register 0 for word access | CSIHnTX0W | <CSIHn_base> + 1008 _H |
| CSIHn | CSIHn transmit data register 0 for half word access | CSIHnTX0H | <CSIHn_base> + 100C _H |
| CSIHn | CSIHn receive data register 0 for word access | CSIHnRX0W | <CSIHn_base> + 1010 _H |
| CSIHn | CSIHn receive data register 0 for half word access | CSIHnRX0H | <CSIHn_base> + 1014 _H |
| CSIHn | CSIHn baud rate setting register 0 | CSIHnBRS0 | <CSIHn_base> + 1068 _H |
| CSIHn | CSIHn baud rate setting register 1 | CSIHnBRS1 | <CSIHn_base> + 106C _H |
| CSIHn | CSIHn baud rate setting register 2 | CSIHnBRS2 | <CSIHn_base> + 1070 _H |
| CSIHn | CSIHn baud rate setting register 3 | CSIHnBRS3 | <CSIHn_base> + 1074 _H |
| CSIH | CSIH DMA select register* ¹ | SELCSIHDMA | <CSIH0_base> + E000 _H |

Note 1. Supported only for CSIH0 and CSIH1.

14.3.2 CSIHnCTL0 — CSIHn Control Register 0

This register controls the operation clock, enables/disables transmission/reception, and enables/disables the memory usage for transmission and/or reception. It forces the stop of communication at the end of the current job.

Access: This register can be read/written in 8- or 1-bit units.

Address: <CSIHn_base> + 0000_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|----------|----------|---|---|---|-----------|----------|
| | CSIHnPWR | CSIHnTXE | CSIHnRXE | — | — | — | CSIHnJOBE | CSIHnMBS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R/W | R/W |

Table 14.13 CSIHnCTL0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 | CSIHnPWR | Controls the operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets CSIH to standby state. Clock supply to internal circuits stops. If CSIHnPWR is cleared to 0 during communication, ongoing communication is immediately aborted. In this case, communication settings must be made from the beginning. |
| 6 | CSIHnTXE | Permits or prohibits transmission. 0: Prohibits transmission. 1: Permits transmission. |
| 5 | CSIHnRXE | Permits or prohibits reception. 0: Prohibits reception. 1: Permits reception. |
| 4 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | CSIHnJOBE | Stops communication at the end of the current job. (Communication ends when data is written to the transmit buffer while CSIHnTXOW.CSIHnEOJ = 1 (job completion).) 0: Communication stop is not requested. 1: Stops communication. This bit can be used to abort an ongoing job. This bit is cleared to 0 automatically. Even if this bit is set to 1, the read value is always 0. In FIFO mode, the pointer must be cleared by setting CSIHnSTCR0.CSIHnPCT = 1 before the next communication is started |
| 0 | CSIHnMBS | Bypasses the memory for transmission and/or reception data. 0: Memory mode CSIH memory is used for transmission and/or reception data. 1: Direct access mode CSIH memory is bypassed. |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers**.

14.3.3 CSIHnCTL1 — CSIHn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, handshake functionality, and job mode. It selects the active output level of each chip select signal and the behavior of the chip select signals after the transfer of the final data.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 0010_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|----------|-----------|------------|-----------|-----------|-----------|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CSIHn PHE | CSIHn CKR | CSIHn SLIT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CSIHn CSL7 | CSIHn CSL6 | CSIHn CSL5 | CSIHn CSL4 | CSIHn CSL3 | CSIHn CSL2 | CSIHn CSL1 | CSIHn CSL0 | CSIHn EDLE | CSIHn JE | CSIHn DCS | CSIHn CSRI | CSIHn LBM | CSIHn SIT | CSIHn HSE | CSIHn SSE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 14.14 CSIHnCTL1 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 31 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | CSIHnPHE | Sets the CPU-controlled priority-based communication function. 0: The CPU-controlled high-priority communication function is disabled. 1: The CPU-controlled high-priority communication function is enabled. To enable the CPU-controlled high-priority communication function, set this bit to 1 and set CSIHnJE = 1. This bit can only be set in transmit-only buffer mode. |
| 17 | CSIHnCKR | CSIHnTSCK clock inversion function 0: The default level of CSIHnTSCK is high 1: The default level of CSIHnTSCK is low For details, see Section 14.3.11, CSIHnCFGx — CSIHn Configuration Register x . |
| 16 | CSIHnSLIT | Selects the timing of interrupt INTCSIHnIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: As soon as the contents of the CSIHnTX0W/H register are transferred to the shift register, an interrupt is generated (this function is activated only in direct access mode and transmit-only buffer mode). For details, see Section 14.4.3, INTCSIHnIC (Communication Status Interrupt) . |
| 15 to 8 | CSIHnCSL[7:0] | Selects the active output level of chip select signal x (CSIHnCSSx). (x = 0 to 7) 0: Chip select signal is active low. 1: Chip select signal is active high. For details, see Section 14.5.3, Chip Selection (CS) Features . |
| 7 | CSIHnEDLE | Enables/disables extended data length (EDL) mode. 0: Disables extended data length mode. 1: Enables extended data length mode. For details, see Section 14.5.8.2, Data Length Greater than 16 Bits . |

Table 14.14 CSIHnCTL1 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 6 | CSIHnJE | <p>Enables/disables job mode.</p> <p>0: Disables job mode. 1: Enables job mode.</p> <p>For details, see Section 14.5.3.3, Job Concept. The CSIHnCTL0.CSIHnJOBE, CSIHnTX0W.CSIHnEOJ, and CSIHnTX0W.CSIHnCIRE bits are enabled only when CSIHnJE = 1. Setting this bit in slave mode is prohibited. In addition, to enable the CPU-controlled high-priority communication function, set this bit to 1 as well as CSIHnPHE = 1.</p> |
| 5 | CSIHnDCS | <p>Enables/disables data consistency check.</p> <p>0: Disables data consistency check. 1: Enables data consistency check.</p> <p>For details, see Section 14.5.12.1, Data Consistency Check.</p> |
| 4 | CSIHnCSRI | <p>Defines chip select signal behavior after last data transfer.</p> <p>0: Chip select signal holds the active level. 1: Chip select signal returns to the inactive level.</p> <p>The last data is judged at the following times.</p> <ul style="list-style-type: none"> • INTCSIHnIC interrupt output timing when the next writing is not made at the end of communication in FIFO mode • INTCSIHnIC interrupt output timing while CSIHnCTL1.CSIHnSLIT is set to 0 in direct access mode |
| 3 | CSIHnLBM | <p>Controls loop-back mode (LBM).</p> <p>0: Deactivates loop-back mode. 1: Activates loop-back mode.</p> <p>For details, see Section 14.5.13, Loop-Back Mode.</p> |
| 2 | CSIHnSIT | <p>Selects interrupt delay mode.</p> <p>0: No delay is generated. 1: Half clock delay is generated for all interrupts.</p> <p>This bit is only valid in master mode. In slave mode, no delay is generated. For details, see Section 14.4.2, Interrupt Delay.</p> |
| 1 | CSIHnHSE | <p>Enables/disables the handshake function.</p> <p>0: Disables the handshake function. 1: Enables the handshake function.</p> <p>For details, refer to Section 14.5.11, Handshake Function.</p> |
| 0 | CSIHnSSE | <p>Enables/disables the slave select function.</p> <p>0: Input signal CSIHTSSI is disabled. 1: Input signal CSIHTSSI is enabled.</p> <p>If the slave select function is not used, this bit must be set to 0 (see also Section 14.5.2, Master/Slave Connections).</p> |

Details about CSIHnCTL1.CSIHnSSE are shown in the following tables.

Table 14.15 Operation of the Slave Select Function during Reception

| CSIHnCTL0.CSIHnRXE | CSIHnCTL1.CSIHnSSE | CSIHTSSI | Receive Operation |
|--------------------|--------------------|----------|-------------------------|
| 0 | — | — | Reception is prohibited |
| 1 | 0 | — | Possible |
| 1 | 1 | 0 | Possible |
| 1 | 1 | 1 | Disabled |

Table 14.16 Operation of the Slave Select Function during Transmission

| CSIHnCTL0. CSIHnTXE | CSIHnCTL1. CSIHnSSE | $\overline{\text{CSIHTSSI}}$ | Transmit Operation |
|------------------------|------------------------|------------------------------|----------------------------|
| 0 | — | — | Transmission is prohibited |
| 1 | 0 | — | Possible |
| 1 | 1 | 0 | Possible |
| 1 | 1 | 1 | Disabled |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.4 CSIHnCTL2 — CSIHn Control Register 2

This register selects operating mode and the reference clock value, and specifies the transfer clock frequency.

For details, see **Section 14.5.5, Transmission Clock Selection**.

Access: This register can be read/written in 16-bit units.

Address: <CSIHn_base> + 0014_H

Value after reset: E000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|-----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | CSIHnPRS[2:0] | | | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 14.17 CSIHnCTL2 Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---------------|--|---|-----------|-----------|---------------------------------------|---|---|---|--------------------|---|---|---|----------------------|---|---|---|----------------------|---|---|---|----------------------|---|---|---|-----------------------|---|---|---|-----------------------|---|---|---|-----------------------|---|---|---|---|
| 15 to 13 | CSIHnPRS[2:0] | These bits select the operation mode and the reference clock value. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>CSIHnPRS2</th> <th>CSIHnPRS1</th> <th>CSIHnPRS0</th> <th>Selection of Reference Clock (PRSOUT)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>PCLK (Master mode)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>PCLK/2 (Master mode)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>PCLK/4 (Master mode)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>PCLK/8 (Master mode)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>PCLK/16 (Master mode)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>PCLK/32 (Master mode)</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>PCLK/64 (Master mode)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>External clock via CSIHnTSCK(in) (Slave mode)</td> </tr> </tbody> </table> | CSIHnPRS2 | CSIHnPRS1 | CSIHnPRS0 | Selection of Reference Clock (PRSOUT) | 0 | 0 | 0 | PCLK (Master mode) | 0 | 0 | 1 | PCLK/2 (Master mode) | 0 | 1 | 0 | PCLK/4 (Master mode) | 0 | 1 | 1 | PCLK/8 (Master mode) | 1 | 0 | 0 | PCLK/16 (Master mode) | 1 | 0 | 1 | PCLK/32 (Master mode) | 1 | 1 | 0 | PCLK/64 (Master mode) | 1 | 1 | 1 | External clock via CSIHnTSCK(in) (Slave mode) |
| CSIHnPRS2 | CSIHnPRS1 | CSIHnPRS0 | Selection of Reference Clock (PRSOUT) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | PCLK (Master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | PCLK/2 (Master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | PCLK/4 (Master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | PCLK/8 (Master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | PCLK/16 (Master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | PCLK/32 (Master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | PCLK/64 (Master mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | External clock via CSIHnTSCK(in) (Slave mode) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

In master mode, the following bits are used to set the transfer clock frequency:

- CSIHnCTL2.CSIHnPRS[2:0],
- CSIHnCFGx.CSIHnBRSS[1:0], and
- CSIHnBRSy.CSIHnBRS[11:0]

In addition, any of the four different transfer clock frequency settings that are specified by the CSIHnBRSy.CSIHnBRS[11:0] bits is selected for each chip select signal. To select the transfer clock frequency setting for each chip select signal, use the CSIHnCFGx.CSIHnBRSS[1:0] bits.

The following table shows the relationship between CSIHnCFGx.CSIHnBRSS[1:0] and CSIHnBRSSy.CSIHnBRS[11:0].

Table 14.18 Relationship Between CSIHnCFGx.CSIHnBRSS[1:0] and CSIHnBRSSy.CSIHnBRS[11:0]

| CSIHnCFGx (x = 0 to 7) CSIHnBRSS1, CSIHnBRSS0 | Transfer Clock Frequency Setting Bit to be Selected |
|---|---|
| 00 | CSIHnBRS0.CSIHnBRS[11:0] |
| 01 | CSIHnBRS1.CSIHnBRS[11:0] |
| 10 | CSIHnBRS2.CSIHnBRS[11:0] |
| 11 | CSIHnBRS3.CSIHnBRS[11:0] |

The following table shows the relationship between the transfer clock frequency setting (CSIHnBRSSy[11:0]) and the transfer clock frequency selected by the CSIHnBRSS[1:0] bits when the CSIHnPRS[2:0] bit value is α .

Table 14.19 Relationship Between CSIHnBRSSy[11:0] and Transfer Clock Frequency

| CSIHnBRSS[11:0] | Transfer Clock Frequency |
|-----------------|--|
| 0 | BRG stopped |
| 1 | $PCLK / (2^\alpha \times 1 \times 2)$ |
| 2 | $PCLK / (2^\alpha \times 2 \times 2)$ |
| 3 | $PCLK / (2^\alpha \times 3 \times 2)$ |
| 4 | $PCLK / (2^\alpha \times 4 \times 2)$ |
| ... | ... |
| 4095 | $PCLK / (2^\alpha \times 4095 \times 2)$ |

When time-out errors are being used in slave mode, the clock selected by this setting is used. When the clock is being used in slave mode, although the CSIHnPRS[2:0] bits are set to 111_B, this has the same effect on the prescaler as the CSIHnPRS[2:0] bits are being set to 000_B. Set the CSIHnBRSSy.CSIHnBRS[11:0] bits to a value other than 000_H to use time-out errors.

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers**.

14.3.5 CSIHnSTR0 — CSIHn Status Register 0

This register indicates the status of CSIH.

Access: This register can only be read in 32-bit units.

Address: <CSIHn_base> + 0004_H

Value after reset: 0000 0010_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|--------------|----|----|----|----|----|---------------|---------------|----|--------------|--------------|--------------|----|-------------|--------------|
| | CSIHnSRP[7:0] | | | | | | | | CSIHnSPF[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CSIHn TMOE | CSIHn OFE | — | — | — | — | — | CSIHn HPST | CSIHn TSF | — | CSIHn FLF | CSIHn EMF | CSIHn DCE | — | CSIHn PE | CSIHn OVE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 14.20 CSIHnSTR0 Register Contents (1/3)

| Bit Position | Bit Name | Function | | | | | | | | | | |
|----------------------|--|--|---------------|-------------|-----------------|--|-----|--|-----------------|--|----------------------|-----------|
| 31 to 24 | CSIHnSRP[7:0] | Indicates the number of received data in FIFO mode. <table border="1"> <thead> <tr> <th>CSIHnSRP[7:0]</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00_H</td> <td>Number of received data packets (0 to 128)</td> </tr> <tr> <td>...</td> <td></td> </tr> <tr> <td>80_H</td> <td></td> </tr> <tr> <td>Other than the above</td> <td>Undefined</td> </tr> </tbody> </table> <p>These bits are cleared by CSIHnSTCR0.CSIHnPCT. In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00_H. In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIHnMCTL2.CSIHnND[7:0].</p> | CSIHnSRP[7:0] | Description | 00 _H | Number of received data packets (0 to 128) | ... | | 80 _H | | Other than the above | Undefined |
| CSIHnSRP[7:0] | Description | | | | | | | | | | | |
| 00 _H | Number of received data packets (0 to 128) | | | | | | | | | | | |
| ... | | | | | | | | | | | | |
| 80 _H | | | | | | | | | | | | |
| Other than the above | Undefined | | | | | | | | | | | |
| 23 to 16 | CSIHnSPF[7:0] | Indicates the number of untransmitted data in FIFO mode. (The number of data written by the CPU is the number of transmission data.) <table border="1"> <thead> <tr> <th>CSIHnSPF[7:0]</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00_H</td> <td>Number of untransmitted data (0 to 128)</td> </tr> <tr> <td>...</td> <td></td> </tr> <tr> <td>80_H</td> <td></td> </tr> <tr> <td>Other than the above</td> <td>Undefined</td> </tr> </tbody> </table> <p>These bits are cleared by CSIHnSTCR0.CSIHnPCT. In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00_H. In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIHnMCTL2.CSIHnND[7:0].</p> | CSIHnSPF[7:0] | Description | 00 _H | Number of untransmitted data (0 to 128) | ... | | 80 _H | | Other than the above | Undefined |
| CSIHnSPF[7:0] | Description | | | | | | | | | | | |
| 00 _H | Number of untransmitted data (0 to 128) | | | | | | | | | | | |
| ... | | | | | | | | | | | | |
| 80 _H | | | | | | | | | | | | |
| Other than the above | Undefined | | | | | | | | | | | |

Table 14.20 CSIHnSTR0 Register Contents (2/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | |
|-----------------------|--|--|---|-----------------|--|-----------------|-------------------------------|---|--------------------|--|----------------------------------|---|-----------------------|-------------------|--------------------------|--|
| 15 | CSIHnTMOE | <p>Time-out error flag in FIFO mode. Indicates whether a time-out error was detected in FIFO mode. 0: No time out error is detected. 1: A time out error is detected. For details, see Section 14.5.12.3, Time-Out Error. This bit is cleared by CSIHnSTCR0.CSIHnTMOEC. However, if this bit is set to 1 by the detection of timeout error and cleared to 0 by CSIHnSTCR0.CSIHnTMOEC simultaneously, When setting to 1 by time-out error detection and clearing to 0 by CSIHnSTCR0.CSIHnTMOEC occur simultaneously, setting to 1 takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.</p> | | | | | | | | | | | | | | |
| 14 | CSIHnOFE | <p>Overflow error flag in FIFO mode. Indicates whether an overflow error was detected in FIFO mode. 0: No overflow error is detected. 1: An overflow error is detected. For details, see Section 14.5.12.4, Overflow Error. This bit is cleared by CSIHnSTCR0.CSIHnOFEC. However, if this bit is set to 1 by the detection of overflow error and cleared to 0 by CSIHnSTCR0.CSIHnOFEC simultaneously, When setting to 1 by overflow error detection and clearing to 0 by CSIHnSTCR0.CSIHnOFEC occur simultaneously, setting to 1 takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. When CSIHnCTL0.CSIHnPWR = 0, if transmission data (129) is written to CSIHnTX0W or CSIHnTX0H, an overflow error occurs.</p> | | | | | | | | | | | | | | |
| 13 to 9 | Reserved | When read, the value after reset is read. | | | | | | | | | | | | | | |
| 8 | CSIHnHPST | <p>Communication priority indication flag 0: Indicates low-priority communication is in progress. 1: Indicates high-priority communication is in progress. This bit always reads 0 if CPU-controlled high-priority communication is disabled (CSIHnPHE = 0).</p> | | | | | | | | | | | | | | |
| 7 | CSIHnTSF | <p>Transfer status flag. 0: Idle state 1: Communication is in progress or being prepared. The timing to set or clear this bit is as follows:</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th rowspan="2">Master Mode</th> <th colspan="2">Timing to Set</th> <th rowspan="2">Timing to Clear</th> </tr> <tr> <th>Direct Access Mode, FIFO Mode</th> <th>Dual Buffer Mode, Transmit-only Buffer Mode</th> </tr> </thead> <tbody> <tr> <td>Transmit-only mode</td> <td rowspan="3">Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)</td> <td rowspan="3">Bit CSIHnMCTL2. CSIHnBTST is set</td> <td rowspan="3">Within a half clock of the last serial clock edge</td> </tr> <tr> <td>Transmit/receive mode</td> </tr> <tr> <td>Receive-only mode</td> </tr> </tbody> </table> | Master Mode | Timing to Set | | Timing to Clear | Direct Access Mode, FIFO Mode | Dual Buffer Mode, Transmit-only Buffer Mode | Transmit-only mode | Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H) | Bit CSIHnMCTL2. CSIHnBTST is set | Within a half clock of the last serial clock edge | Transmit/receive mode | Receive-only mode | | |
| Master Mode | Timing to Set | | | Timing to Clear | | | | | | | | | | | | |
| | Direct Access Mode, FIFO Mode | Dual Buffer Mode, Transmit-only Buffer Mode | | | | | | | | | | | | | | |
| Transmit-only mode | Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H) | Bit CSIHnMCTL2. CSIHnBTST is set | Within a half clock of the last serial clock edge | | | | | | | | | | | | | |
| Transmit/receive mode | | | | | | | | | | | | | | | | |
| Receive-only mode | | | | | | | | | | | | | | | | |
| | | <table border="1" style="margin-left: 20px;"> <thead> <tr> <th rowspan="2">Slave Mode</th> <th colspan="2">Timing to Set</th> <th rowspan="2">Timing to Clear</th> </tr> <tr> <th>Direct Access Mode, FIFO Mode</th> <th>Dual Buffer Mode, Transmit-Only Buffer Mode</th> </tr> </thead> <tbody> <tr> <td>Transmit-only mode</td> <td rowspan="2">Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)</td> <td rowspan="2">Bit CSIHnMCTL2. CSIHnBTST is set</td> <td rowspan="3">Within a half clock of the last serial clock edge</td> </tr> <tr> <td>Transmit/receive mode</td> </tr> <tr> <td>Receive-only mode</td> <td>Input timing of CSIHnTSC</td> <td></td> </tr> </tbody> </table> | Slave Mode | Timing to Set | | Timing to Clear | Direct Access Mode, FIFO Mode | Dual Buffer Mode, Transmit-Only Buffer Mode | Transmit-only mode | Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H) | Bit CSIHnMCTL2. CSIHnBTST is set | Within a half clock of the last serial clock edge | Transmit/receive mode | Receive-only mode | Input timing of CSIHnTSC | |
| Slave Mode | Timing to Set | | | Timing to Clear | | | | | | | | | | | | |
| | Direct Access Mode, FIFO Mode | Dual Buffer Mode, Transmit-Only Buffer Mode | | | | | | | | | | | | | | |
| Transmit-only mode | Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H) | Bit CSIHnMCTL2. CSIHnBTST is set | Within a half clock of the last serial clock edge | | | | | | | | | | | | | |
| Transmit/receive mode | | | | | | | | | | | | | | | | |
| Receive-only mode | Input timing of CSIHnTSC | | | | | | | | | | | | | | | |
| 6 | Reserved | When read, the value after reset is read. | | | | | | | | | | | | | | |

Table 14.20 CSIHnSTR0 Register Contents (3/3)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 5 | CSIHnFLF | A flag indicating that the buffer is full in FIFO mode. 0: FIFO buffer is not full. 1: FIFO buffer is full. This bit is cleared by CSIHnSTCR0.CSIHnPCT. The FIFO buffer might be filled with untransmitted data or received data. |
| 4 | CSIHnEMF | A flag indicating that the buffer is empty in FIFO mode. 0: FIFO buffer is not empty. 1: FIFO buffer is empty. This bit is set to 1 by CSIHnSTCR0.CSIHnPCT. This bit is set to 1 when CSIHnSTR0.CSIHnSRP[7:0] + CSIHnSTR0.CSIHnSPF[7:0] = 00 _H . |
| 3 | CSIHnDCE | Data consistency check error flag. 0: No data consistency check error is detected. 1: Data consistency check error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnDCEC. However, if this bit is set to 1 by the detection of a data consistency check error and cleared to 0 by CSIHnSTCR0.CSIHnDCEC simultaneously, setting to 1 due to the detection of a data consistency check error is prioritized. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. |
| 2 | Reserved | When read, the value after reset is read. |
| 1 | CSIHnPE | Parity error flag 0: No parity error is detected. 1: Parity error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnPEC. However, if this bit is set to 1 by the detection of parity error and cleared to 0 by CSIHnSTCR0.CSIHnPEC simultaneously, setting to 1 due to the detection of a parity error is prioritized. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. |
| 0 | CSIHnOVE | Overrun error flag (Fixed to 0 in dual buffer mode). 0: No overrun error is detected. 1: Overrun error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnOVEC. However, if this bit is set to 1 by the detection of overrun error and cleared to 0 by CSIHnSTCR0.CSIHnOVEC simultaneously, setting to 1 due to the detection of an overrun error is prioritized. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. |

Table 14.21 Behaviors in Various Memory Modes

| Bit Name | Bit Position | Direct Access Mode | FIFO Mode | Transmit-only Buffer Mode | Dual Buffer Mode |
|---------------|--------------|---|---|--|------------------|
| CSIHnSRP[7:0] | 31 to 24 | Fixed to 0 | Number of received data | Fixed to 0 | Fixed to 0 |
| CSIHnSPF[7:0] | 23 to 16 | Fixed to 0 | Number of untransmitted data | Fixed to 0 | Fixed to 0 |
| CSIHnTMOE | 15 | Fixed to 0 | Time-out error flag 0: No error is detected. 1: An error is detected. | Fixed to 0 | Fixed to 0 |
| CSIHnOFE | 14 | Fixed to 0 | Overflow error flag 0: No error is detected. 1: An error is detected. | Fixed to 0 | Fixed to 0 |
| CSIHnTSF | 7 | 0: Idle state 1: Transmission is in progress or being prepared | | | |
| CSIHnFLF | 5 | Fixed to 0 | FIFO full flag 0: FIFO is not full. 1: FIFO is full. | Fixed to 0 | Fixed to 0 |
| CSIHnEMF | 4 | Fixed to 1 | FIFO empty flag 0: FIFO is not empty. 1: FIFO is empty. | Fixed to 1 | Fixed to 1 |
| CSIHnDCE | 3 | 0: No error is detected. 1: An error is detected. | | | |
| CSIHnPE | 1 | 0: No error is detected. 1: An error is detected. | | | |
| CSIHnOVE | 0 | 0: No error is detected. 1: An error is detected. | 0: No error is detected. 1: An error is detected. | 0: No error is detected. 1: An error is detected. | Fixed to 0 |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.6 CSIHnSTCR0 — CSIHn Status Clear Register 0

This register clears the status flags of the CSIHnSTR0 status register.

Access: This register can be read/written in 16-bit units.
When read, the value 0000_H is always returned.

Address: <CSIHn_base> + 0008_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|-----------|----|----|----|----|---|----------|---|---|---|---|-----------|---|----------|-----------|
| | CSIHnTMOEC | CSIHnOFEC | — | — | — | — | — | CSIHnPCT | — | — | — | — | CSIHnDCEC | — | CSIHnPEC | CSIHnOVEC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R/W | R | R | R | R | R/W | R | R/W | R/W |

Table 14.22 CSIHnSTCR0 Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | |
|---------------------------|-------------------------|--|---------------------|------------|---------------------------|-------------------------|--------------------------|-------------------------|--------------------------|--------------------|--|--------------------|
| 15 | CSIHnTMOEC | Controls the time-out error flag clear command. 0: No operation. 1: Clears the time out error flag (CSIHnSTR0.CSIHnTMOE). The read value is always 0. | | | | | | | | | | |
| 14 | CSIHnOFEC | Controls the overflow error flag clear command. 0: No operation. 1: Clears the overflow error flag (CSIHnSTR0.CSIHnOFFE). The read value is always 0. | | | | | | | | | | |
| 13 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | |
| 8 | CSIHnPCT | Controls the FIFO pointer clear command. 0: No operation. 1: Clears the following FIFO buffer pointers (in FIFO mode, dual buffer mode, and transmit-only buffer mode) and the status bits below. The read value is always 0. | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>FIFO Buffer Pointer</th> <th>Status Bit</th> </tr> </thead> <tbody> <tr> <td>CSIHnMRWP0.CSIHnTRWA[6:0]</td> <td>CSIHnSTR0.CSIHnSPF[7:0]</td> </tr> <tr> <td>CSIHnMRWP0.CSIHnRRA[6:0]</td> <td>CSIHnSTR0.CSIHnSRP[7:0]</td> </tr> <tr> <td>CSIHnMCTL2.CSIHnSOP[6:0]</td> <td>CSIHnSTR0.CSIHnFLF</td> </tr> <tr> <td></td> <td>CSIHnSTR0.CSIHnTSF</td> </tr> </tbody> </table> | FIFO Buffer Pointer | Status Bit | CSIHnMRWP0.CSIHnTRWA[6:0] | CSIHnSTR0.CSIHnSPF[7:0] | CSIHnMRWP0.CSIHnRRA[6:0] | CSIHnSTR0.CSIHnSRP[7:0] | CSIHnMCTL2.CSIHnSOP[6:0] | CSIHnSTR0.CSIHnFLF | | CSIHnSTR0.CSIHnTSF |
| FIFO Buffer Pointer | Status Bit | | | | | | | | | | | |
| CSIHnMRWP0.CSIHnTRWA[6:0] | CSIHnSTR0.CSIHnSPF[7:0] | | | | | | | | | | | |
| CSIHnMRWP0.CSIHnRRA[6:0] | CSIHnSTR0.CSIHnSRP[7:0] | | | | | | | | | | | |
| CSIHnMCTL2.CSIHnSOP[6:0] | CSIHnSTR0.CSIHnFLF | | | | | | | | | | | |
| | CSIHnSTR0.CSIHnTSF | | | | | | | | | | | |
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | |
| 3 | CSIHnDCEC | Controls the data consistency check error flag clear command. 0: No operation. 1: Clears the data consistency check error flag (CSIHnSTR0.CSIHnDCE). The read value is always 0. | | | | | | | | | | |
| 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | |
| 1 | CSIHnPEC | Controls the parity error flag clear command. 0: No operation. 1: Clears the parity error flag (CSIHnSTR0.CSIHnPE). The read value is always 0. | | | | | | | | | | |
| 0 | CSIHnOVEC | Controls the overrun error flag clear command. 0: No operation. 1: Clears the overrun error flag (CSIHnSTR0.CSIHnOVE). The read value is always 0. | | | | | | | | | | |

CAUTION

For the setting of this register, refer to **Table 14.34, Notes on Setting Registers**.

14.3.7 CSIHnMCTL0 — CSIHn Memory Control Register 0

This register selects the memory mode and the time-out setting.

Access: This register can be read/written in 16-bit units.

Address: <CSIHn_base> + 1040_H

Value after reset: 001F_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|---------------|-----|---|---|---|--------------|-----|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | CSIHnMMS[1:0] | | — | — | — | CSIHnTO[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 14.23 CSIHnMCTL0 Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------------|--|---|--------------|-------------|--------------------|-------------------------|--------------------|---|--------------------|---|------------------|---|--------------------|--|---|---|------------|
| 15 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 9, 8 | CSIHnMMS [1:0] | <p>Selects the memory mode.</p> <table border="1"> <thead> <tr> <th>CSIHnMMS1</th> <th>CSIHnMMS0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>FIFO mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>Dual buffer mode</td> </tr> <tr> <td>1</td> <td>0</td> <td>Transmit-only buffer mode</td> </tr> <tr> <td>1</td> <td>1</td> <td>Prohibited</td> </tr> </tbody> </table> <p>After a change of the memory mode, the respective buffer pointers must be cleared by setting the CSIHnSTCR0.CSIHnPCT bit to 1. In direct access mode, the setting of these bits is ignored.</p> | CSIHnMMS1 | CSIHnMMS0 | Description | 0 | 0 | FIFO mode | 0 | 1 | Dual buffer mode | 1 | 0 | Transmit-only buffer mode | 1 | 1 | Prohibited |
| CSIHnMMS1 | CSIHnMMS0 | Description | | | | | | | | | | | | | | | |
| 0 | 0 | FIFO mode | | | | | | | | | | | | | | | |
| 0 | 1 | Dual buffer mode | | | | | | | | | | | | | | | |
| 1 | 0 | Transmit-only buffer mode | | | | | | | | | | | | | | | |
| 1 | 1 | Prohibited | | | | | | | | | | | | | | | |
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 4 to 0 | CSIHnTO[4:0] | <p>Selects the time-out setting in FIFO mode.</p> <table border="1"> <thead> <tr> <th>CSIHnTO[4:0]</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00000_B</td> <td>No time-out is detected</td> </tr> <tr> <td>00001_B</td> <td>Time-out is (1 × 8 × BRG output clocks)</td> </tr> <tr> <td>00010_B</td> <td>Time-out is (2 × 8 × BRG output clocks)</td> </tr> <tr> <td>...</td> <td></td> </tr> <tr> <td>11111_B</td> <td>Time-out is (31 × 8 × BRG output clocks)</td> </tr> </tbody> </table> <p>CAUTION</p> <p>Changing the time-out setting is only permitted when CSIHnCTL0.CSIHnPWR = 0. Set the CSIHnTO[4:0] bits to 00000_B in modes other than FIFO mode (direct access mode, dual buffer mode, and transmit-only buffer mode). For details about time-out detection, see also Section 14.5.12.3, Time-Out Error.</p> | CSIHnTO[4:0] | Description | 00000 _B | No time-out is detected | 00001 _B | Time-out is (1 × 8 × BRG output clocks) | 00010 _B | Time-out is (2 × 8 × BRG output clocks) | ... | | 11111 _B | Time-out is (31 × 8 × BRG output clocks) | | | |
| CSIHnTO[4:0] | Description | | | | | | | | | | | | | | | | |
| 00000 _B | No time-out is detected | | | | | | | | | | | | | | | | |
| 00001 _B | Time-out is (1 × 8 × BRG output clocks) | | | | | | | | | | | | | | | | |
| 00010 _B | Time-out is (2 × 8 × BRG output clocks) | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | |
| 11111 _B | Time-out is (31 × 8 × BRG output clocks) | | | | | | | | | | | | | | | | |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers**.

14.3.8 CSIHnMCTL1 — CSIHn Memory Control Register 1

This register selects the conditions to generate the interrupt requests, INTCSIHnIC and INTCSIHnIR in FIFO mode.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|---------------|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | CSIHnFES[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | | | | | | | | | CSIHnFFS[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 14.24 CSIHnMCTL1 Register Contents

| Bit position | Bit name | Function |
|--------------|---------------|--|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | CSIHnFES[6:0] | Selects the conditions to generate the INTCSIHnIC interrupt (transmit data empty) in FIFO mode. When the number of untransmitted data to be transmitted in FIFO (checked by the CSIHnSTR0.CSIHnSPF[7:0] bit) and CSIHnMCTL1.CSIHnFES[6:0] match, the INTCSIHnIC interrupt request is generated. |
| 15 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | CSIHnFFS[6:0] | Selects the conditions to generate the INTCSIHnIR interrupt (receive data full) in FIFO mode. When the number of received data in FIFO (checked by the CSIHnSTR0.CSIHnSRP[7:0] bit) and (128 - CSIHnMCTL1.CSIHnFFS[6:0]) match, the INTCSIHnIR interrupt request is generated. |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.9 CSIHnMCTL2 — CSIHn Memory Control Register 2

This register controls the operation of the memory in dual buffer or transmit-only buffer mode and triggers to start communication in these buffer modes.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1004_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|--------------|---------------|-----|-----|-----|-----|-----|-----|
| | CSIHn BTST | — | — | — | — | — | — | — | CSIHnND[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | CSIHnSOP[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 14.25 CSIHnMCTL2 Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------|------------------------|--|---------------|--------------------|---------------------------|-----------|--------------------|-----------------|-----------------|-----------------|--------------|--------------|-----------------|-----------------|-----------------|--------------|--------------|-----|-----|-----|--------------|--------------|-----------------|------------------|------------------|--------------|--------------|-----------------|------------------|------------------|--------------|--------------|-----|------------|-----|--------------|--------------|-----------------|------------|-------------------|--------------|--------------|-----------------|------------|-------------------|--------------|--------------|----------------------|------------------------|--|--|--|
| 31 | CSIHnBTST | Provides a start trigger for buffer transfer. 0: No operation. 1: Issues the start transfer command. The read value is always 0. CAUTION This bit can only be used in dual buffer mode and transmit-only buffer mode. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 30 to 24 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 23 to 16 | CSIHnND[7:0] | Specifies the number of data for each memory mode. The read value indicates the number of remaining communication data. <table border="1"> <thead> <tr> <th>CSIHn ND[7:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <tr> <td>00_H</td> <td>Transmit 0 data</td> <td>Transmit 0 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>01_H</td> <td>Transmit 1 data</td> <td>Transmit 1 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>3F_H</td> <td>Transmit 63 data</td> <td>Transmit 63 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>40_H</td> <td>Transmit 64 data</td> <td>Transmit 64 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>Prohibited</td> <td>...</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>7F_H</td> <td>Prohibited</td> <td>Transmit 127 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>80_H</td> <td>Prohibited</td> <td>Transmit 128 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>Other than the above</td> <td colspan="4">Setting is prohibited.</td> </tr> </tbody> </table> The values are automatically decremented after data transfer (not decremented in direct access mode). | CSIHn ND[7:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | 00 _H | Transmit 0 data | Transmit 0 data | No influence | No influence | 01 _H | Transmit 1 data | Transmit 1 data | No influence | No influence | ... | ... | ... | No influence | No influence | 3F _H | Transmit 63 data | Transmit 63 data | No influence | No influence | 40 _H | Transmit 64 data | Transmit 64 data | No influence | No influence | ... | Prohibited | ... | No influence | No influence | 7F _H | Prohibited | Transmit 127 data | No influence | No influence | 80 _H | Prohibited | Transmit 128 data | No influence | No influence | Other than the above | Setting is prohibited. | | | |
| CSIHn ND[7:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 _H | Transmit 0 data | Transmit 0 data | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 _H | Transmit 1 data | Transmit 1 data | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | ... | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3F _H | Transmit 63 data | Transmit 63 data | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 40 _H | Transmit 64 data | Transmit 64 data | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | Prohibited | ... | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7F _H | Prohibited | Transmit 127 data | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 80 _H | Prohibited | Transmit 128 data | No influence | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Other than the above | Setting is prohibited. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 14.25 CSIHnMCTL2 Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|-------------------|--|-------------------|--------------------|---------------------------|-----------|--------------------|-----------------|-------------------|-------------------|-------------------|--------------|-----------------|-------------------|-------------------|-------------------|--------------|-----|-----|-----|-----|--------------|-----------------|-------------------|-------------------|-------------------|--------------|-----------------|------------|-------------------|-------------------|--------------|-----|------------|-----|-----|--------------|-----------------|------------|-------------------|-------------------|--------------|
| 6 to 0 | CSIHnSOP[6:0] | Selects the pointer of the data to be transmitted. If communication is forced to stop by setting CSIHnCTL0.CSIHnPWR to 0 or CSIHnSTCR0.CSIHnPCT to 1, these bits are cleared by the hardware. In FIFO mode, these bits indicate the transmitted address. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>CSIHn SOP[6:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <tr> <td>00_H</td> <td>0000_H</td> <td>0000_H</td> <td>0000_H</td> <td>No influence</td> </tr> <tr> <td>01_H</td> <td>0004_H</td> <td>0004_H</td> <td>0004_H</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>No influence</td> </tr> <tr> <td>3F_H</td> <td>00FC_H</td> <td>00FC_H</td> <td>00FC_H</td> <td>No influence</td> </tr> <tr> <td>40_H</td> <td>Prohibited</td> <td>0100_H</td> <td>0100_H</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>Prohibited</td> <td>...</td> <td>...</td> <td>No influence</td> </tr> <tr> <td>7F_H</td> <td>Prohibited</td> <td>01FC_H</td> <td>01FC_H</td> <td>No influence</td> </tr> </tbody> </table> | CSIHn SOP[6:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | 00 _H | 0000 _H | 0000 _H | 0000 _H | No influence | 01 _H | 0004 _H | 0004 _H | 0004 _H | No influence | ... | ... | ... | ... | No influence | 3F _H | 00FC _H | 00FC _H | 00FC _H | No influence | 40 _H | Prohibited | 0100 _H | 0100 _H | No influence | ... | Prohibited | ... | ... | No influence | 7F _H | Prohibited | 01FC _H | 01FC _H | No influence |
| CSIHn SOP[6:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 _H | 0000 _H | 0000 _H | 0000 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 _H | 0004 _H | 0004 _H | 0004 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | ... | ... | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3F _H | 00FC _H | 00FC _H | 00FC _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 40 _H | Prohibited | 0100 _H | 0100 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | Prohibited | ... | ... | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7F _H | Prohibited | 01FC _H | 01FC _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

After the data transfer, the value increments automatically.

CAUTION

In direct access mode, these bits are not incremented.

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers**.

14.3.10 CSIHnMRWP0 — CSIHn Memory Read/Write Pointer Register 0

This register sets the pointers for reading from and writing to the dual or transmit-only buffer.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1018_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----------------|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | CSIHnRRA[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | CSIHnTRWA[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 14.26 CSIHnMRWP0 Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|-------------------|--|-------------------|--------------------|---------------------------|-----------|--------------------|-----------------|-------------------|--------------|-------------------|--------------|-----------------|-------------------|--------------|-------------------|--------------|-----|-----|--------------|-----|--------------|-----------------|-------------------|--------------|-------------------|--------------|-----------------|------------|--------------|-------------------|--------------|-----|------------|--------------|-----|--------------|-----------------|------------|--------------|-------------------|--------------|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 to 16 | CSIHnRRA[6:0] | <p>Selects the read pointer of the receive buffer.</p> <table border="1"> <thead> <tr> <th>CSIHn RRA[6:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <tr> <td>00_H</td> <td>0000_H</td> <td>No influence</td> <td>0000_H</td> <td>No influence</td> </tr> <tr> <td>01_H</td> <td>0004_H</td> <td>No influence</td> <td>0004_H</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>...</td> <td>No influence</td> <td>...</td> <td>No influence</td> </tr> <tr> <td>3F_H</td> <td>00FC_H</td> <td>No influence</td> <td>00FC_H</td> <td>No influence</td> </tr> <tr> <td>40_H</td> <td>Prohibited</td> <td>No influence</td> <td>0100_H</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>Prohibited</td> <td>No influence</td> <td>...</td> <td>No influence</td> </tr> <tr> <td>7F_H</td> <td>Prohibited</td> <td>No influence</td> <td>01FC_H</td> <td>No influence</td> </tr> </tbody> </table> <p>These bits are automatically incremented when received data is read. When CSIHnRRA[6:0] is 7F_H and received data is read, CSIHnRRA[6:0] is 00_H. If an overrun error is generated while reading the CSIHnRX0W or CSIHnRX0H register, the read pointer is not incremented. These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1. In direct access mode and transmit-only buffer mode, these bits are not incremented. To perform write access in transmit-only buffer mode, set these bits to 0000_H. In FIFO mode, these bits indicate the read address of the received data.</p> | CSIHn RRA[6:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | 00 _H | 0000 _H | No influence | 0000 _H | No influence | 01 _H | 0004 _H | No influence | 0004 _H | No influence | ... | ... | No influence | ... | No influence | 3F _H | 00FC _H | No influence | 00FC _H | No influence | 40 _H | Prohibited | No influence | 0100 _H | No influence | ... | Prohibited | No influence | ... | No influence | 7F _H | Prohibited | No influence | 01FC _H | No influence |
| CSIHn RRA[6:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 _H | 0000 _H | No influence | 0000 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 _H | 0004 _H | No influence | 0004 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | No influence | ... | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3F _H | 00FC _H | No influence | 00FC _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 40 _H | Prohibited | No influence | 0100 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | Prohibited | No influence | ... | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7F _H | Prohibited | No influence | 01FC _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 14.26 CSIHnMRWP0 Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|---------------------|---|--------------------|-----------------------|------------------------------|-----------|-----------------------|-----------------|-------------------|-------------------|-------------------|--------------|-----------------|-------------------|-------------------|-------------------|--------------|-----|-----|-----|-----|--------------|-----------------|-------------------|-------------------|-------------------|--------------|-----------------|------------|-------------------|-------------------|--------------|-----|------------|-----|-----|--------------|-----------------|------------|-------------------|-------------------|--------------|
| 6 to 0 | CSIHnTRWA [6:0] | Selects the read/write pointer of the transmit buffer. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>CSIHn TRWA[6:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <tr> <td>00_H</td> <td>0000_H</td> <td>0000_H</td> <td>0000_H</td> <td>No influence</td> </tr> <tr> <td>01_H</td> <td>0004_H</td> <td>0004_H</td> <td>0004_H</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>No influence</td> </tr> <tr> <td>3F_H</td> <td>00FC_H</td> <td>00FC_H</td> <td>00FC_H</td> <td>No influence</td> </tr> <tr> <td>40_H</td> <td>Prohibited</td> <td>0100_H</td> <td>0100_H</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>Prohibited</td> <td>...</td> <td>...</td> <td>No influence</td> </tr> <tr> <td>7F_H</td> <td>Prohibited</td> <td>01FC_H</td> <td>01FC_H</td> <td>No influence</td> </tr> </tbody> </table> | CSIHn TRWA[6:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | 00 _H | 0000 _H | 0000 _H | 0000 _H | No influence | 01 _H | 0004 _H | 0004 _H | 0004 _H | No influence | ... | ... | ... | ... | No influence | 3F _H | 00FC _H | 00FC _H | 00FC _H | No influence | 40 _H | Prohibited | 0100 _H | 0100 _H | No influence | ... | Prohibited | ... | ... | No influence | 7F _H | Prohibited | 01FC _H | 01FC _H | No influence |
| CSIHn TRWA[6:0] | Dual Buffer Mode | Transmit-only Buffer Mode | FIFO Mode | Direct Access Mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 _H | 0000 _H | 0000 _H | 0000 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 _H | 0004 _H | 0004 _H | 0004 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | ... | ... | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3F _H | 00FC _H | 00FC _H | 00FC _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 40 _H | Prohibited | 0100 _H | 0100 _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | Prohibited | ... | ... | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7F _H | Prohibited | 01FC _H | 01FC _H | No influence | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

These bits are automatically incremented when the transmission data is written or read.

When CSIHnTRWA[6:0] is 7F_H and received data is read, CSIHnTRWA[6:0] is 00_H.

These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.

In direct access mode, these bits are not incremented.

In FIFO mode, these bits indicate the read/write address of transmission data.

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.11 CSIHnCFGx — CSIHn Configuration Register x

These eight registers configure the chip select signal CSIHnCSSx baud rate, parity, data length, recessive configuration for broadcasting, serial data direction, clock phase, data phase, idle enforcement configuration, idle time, hold time, inter-data time and setup for each chip select signal, CSIHnCSSx.

Slave mode

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is effective.

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock and data phase

In slave mode, set all bits other than above in the CSIHnCFG0 register, and the CSIHnCFG1 to CSIHnCFG7 registers to 0.

Access: This register can be read/written in 32-bit units.

Address: CSIHnCFG0: <CSIHn_base> + 1044_H
 CSIHnCFG1: <CSIHn_base> + 1048_H
 CSIHnCFG2: <CSIHn_base> + 104C_H
 CSIHnCFG3: <CSIHn_base> + 1050_H
 CSIHnCFG4: <CSIHn_base> + 1054_H
 CSIHnCFG5: <CSIHn_base> + 1058_H
 CSIHnCFG6: <CSIHn_base> + 105C_H
 CSIHnCFG7: <CSIHn_base> + 1060_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------------|---------------|---------------|-----|----------------|-----|-----|-----|---------------|-----|-----|-----|---------------|---------------|---------------|---------------|
| | CSIHnBRSSx [1:0] | | CSIHnPSx[1:0] | | CSIHnDLSx[3:0] | | | | — | — | — | — | CSIHn RCBx | CSIHn DIRx | CSIHn CKPx | CSIHn DAPx |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CSIHn IDLx | CSIHnIDx[2:0] | | | CSIHnHDx[3:0] | | | | CSIHnINx[3:0] | | | | CSIHnSPx[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 14.27 CSIHnCFGx Register Contents (1/4)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | |
|-------------------|---------------------|---|---------------------------------------|-------------|--------------------------------------|-----------|-------------------|---|--------------------------|-----------------------|---|-----|----------------------------|---|---|---|---|----------------------------|---|---|------------------|-----------------------------|
| 31, 30 | CSIHnBRSSx [1:0] | <p>These bits select a baud rate setting register (CSIHnBRSy).</p> <table border="1"> <thead> <tr> <th>CSIHnBRSSx1</th> <th>CSIHnBRSSx0</th> <th>Baud Rate Setting Register Selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>The transfer clock frequency is set according to the CSIHnBRS0 setting.</td> </tr> <tr> <td>0</td> <td>1</td> <td>The transfer clock frequency is set according to the CSIHnBRS1 setting.</td> </tr> <tr> <td>1</td> <td>0</td> <td>The transfer clock frequency is set according to the CSIHnBRS2 setting.</td> </tr> <tr> <td>1</td> <td>1</td> <td>The transfer clock frequency is set according to the CSIHnBRS3 setting.</td> </tr> </tbody> </table> <p>In accordance with the setting of CSIHnCTL2.CSIHnPRS[2:0], the maximum value for setting the transfer clock frequency must be as below. Master mode: PCLK/4 Slave mode: PCLK/6</p> | CSIHnBRSSx1 | CSIHnBRSSx0 | Baud Rate Setting Register Selection | 0 | 0 | The transfer clock frequency is set according to the CSIHnBRS0 setting. | 0 | 1 | The transfer clock frequency is set according to the CSIHnBRS1 setting. | 1 | 0 | The transfer clock frequency is set according to the CSIHnBRS2 setting. | 1 | 1 | The transfer clock frequency is set according to the CSIHnBRS3 setting. | | | | | |
| CSIHnBRSSx1 | CSIHnBRSSx0 | Baud Rate Setting Register Selection | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | The transfer clock frequency is set according to the CSIHnBRS0 setting. | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | The transfer clock frequency is set according to the CSIHnBRS1 setting. | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | The transfer clock frequency is set according to the CSIHnBRS2 setting. | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | The transfer clock frequency is set according to the CSIHnBRS3 setting. | | | | | | | | | | | | | | | | | | | | |
| 29, 28 | CSIHnPSx[1:0] | <p>Selects the parity for transmitting or receiving chip select signal x.</p> <table border="1"> <thead> <tr> <th>CSIHnPSx1</th> <th>CSIHnPSx0</th> <th>Transmission</th> <th>Reception</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No parity is transmitted</td> <td>No parity is expected</td> </tr> <tr> <td>0</td> <td>1</td> <td>Adds parity bit fixed to 0</td> <td>Parity bit is expected but not judged</td> </tr> <tr> <td>1</td> <td>0</td> <td>Adds odd parity only</td> <td>Odd parity bit is expected</td> </tr> <tr> <td>1</td> <td>1</td> <td>Adds even parity</td> <td>Even parity bit is expected</td> </tr> </tbody> </table> | CSIHnPSx1 | CSIHnPSx0 | Transmission | Reception | 0 | 0 | No parity is transmitted | No parity is expected | 0 | 1 | Adds parity bit fixed to 0 | Parity bit is expected but not judged | 1 | 0 | Adds odd parity only | Odd parity bit is expected | 1 | 1 | Adds even parity | Even parity bit is expected |
| CSIHnPSx1 | CSIHnPSx0 | Transmission | Reception | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | No parity is transmitted | No parity is expected | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Adds parity bit fixed to 0 | Parity bit is expected but not judged | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Adds odd parity only | Odd parity bit is expected | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Adds even parity | Even parity bit is expected | | | | | | | | | | | | | | | | | | | |
| 27 to 24 | CSIHnDLSx [3:0] | <p>Selects the data length for chip select signal x.</p> <table border="1"> <thead> <tr> <th>CSIHnDLSx[3:0]</th> <th>Data Length</th> </tr> </thead> <tbody> <tr> <td>0000_B</td> <td>16 bits</td> </tr> <tr> <td>0001_B</td> <td>1 bit</td> </tr> <tr> <td>0010_B</td> <td>2 bits</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>1111_B</td> <td>15 bits</td> </tr> </tbody> </table> <p>CAUTION</p> <p>When CSIHnTX0W.CSIHnEDL = 1, the setting of this bit is invalid (the data length is 16 bits). When CSIHnTX0W.CSIHnEDL = 0, the setting of this bit is valid. Setting the Data Length to "1 bit" is only allowed if the previous transmit data is 16 bits with CSIHnEDL=1.</p> | CSIHnDLSx[3:0] | Data Length | 0000 _B | 16 bits | 0001 _B | 1 bit | 0010 _B | 2 bits | ... | ... | 1111 _B | 15 bits | | | | | | | | |
| CSIHnDLSx[3:0] | Data Length | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | 16 bits | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | 1 bit | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | 2 bits | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | 15 bits | | | | | | | | | | | | | | | | | | | | | |
| 23 to 20 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | |
| 19 | CSIHnRCBx | <p>Selects the recessive configuration for broadcasting for chip select signal x: 0: Dominant (higher priority) 1: Recessive (lower priority) For details, see Section 14.5.3.1, Configuration Registers</p> | | | | | | | | | | | | | | | | | | | | |
| 18 | CSIHnDIRx | <p>Selects the serial data direction of chip select signal x. 0: Data is transmitted/received with MSB first. 1: Data is transmitted/received with LSB first. For details, refer to Section 14.5.9, Serial Data Direction Selection.</p> | | | | | | | | | | | | | | | | | | | | |

Table 14.27 CSIHnCFGx Register Contents (2/4)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--|-----------|---|-----------|-----------|--------------------------------|---|---|--|---|---|--|---|---|--------------------|---|---|--|
| 17 | CSIHnCKPx | CSIHnCKPx: Clock phase selection bit | | | | | | | | | | | | | | | |
| 16 | CSIHnDAPx | CSIHnDAPx: Data phase selection bit <ul style="list-style-type: none"> CSIHnCTL1.CSIHnCKR = 0 | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>CSIHnCKPx</th> <th>CSIHnDAPx</th> <th>Clock and Data Phase Selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td> </td> </tr> <tr> <td>0</td> <td>1</td> <td> </td> </tr> <tr> <td>1</td> <td>0</td> <td> </td> </tr> <tr> <td>1</td> <td>1</td> <td> </td> </tr> </tbody> </table> | | | CSIHnCKPx | CSIHnDAPx | Clock and Data Phase Selection | 0 | 0 | | 0 | 1 | | 1 | 0 | | 1 | 1 | |
| CSIHnCKPx | CSIHnDAPx | Clock and Data Phase Selection | | | | | | | | | | | | | | | |
| 0 | 0 | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | | | | | | |
| <ul style="list-style-type: none"> CSIHnCTL1.CSIHnCKR = 1 | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>CSIHnCKPx</th> <th>CSIHnDAPx</th> <th>Clock and Data Phase Selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td> </td> </tr> <tr> <td>0</td> <td>1</td> <td> </td> </tr> <tr> <td>1</td> <td>x</td> <td>Setting prohibited</td> </tr> </tbody> </table> | | | CSIHnCKPx | CSIHnDAPx | Clock and Data Phase Selection | 0 | 0 | | 0 | 1 | | 1 | x | Setting prohibited | | | |
| CSIHnCKPx | CSIHnDAPx | Clock and Data Phase Selection | | | | | | | | | | | | | | | |
| 0 | 0 | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | x | Setting prohibited | | | | | | | | | | | | | | | |
| 15 | CSIHnIDLx | <p>Selects the idle enforcement configuration for chip select x:</p> <p>0: If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are different, an idle state is inserted between two transfers. If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are the same, an idle state is not inserted between two transfers.</p> <p>1: Regardless of the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers, an idle state is inserted between two transfers.</p> <p>This bit is only available in master mode.</p> <p>For details about the enforced idle state, see Section 14.5.15, Enforced Chip Select Idle Setting.</p> | | | | | | | | | | | | | | | |

Table 14.27 CSIHnCFGx Register Contents (3/4)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---------------------------------------|--|----------------|---------------------------------------|---------------------------------------|-------------------------------------|-------------------------------------|----------------------------|-------------------|-------------------------------|------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------|-------------------------------|------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-----------------------------|
| 14 to 12 | CSIHnIDx[2:0] | Selects the idle time for chip select signal x. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>CSIHnIDx [2:0]</th> <th>Idle Time</th> </tr> </thead> <tbody> <tr> <td>000_B</td> <td>0.5 transmission clock a half cycle</td> </tr> <tr> <td>001_B</td> <td>1 transmission clock cycle</td> </tr> <tr> <td>010_B</td> <td>1.5 transmission clock cycles</td> </tr> <tr> <td>011_B</td> <td>2.5 transmission clock cycles</td> </tr> <tr> <td>100_B</td> <td>3.5 transmission clock cycles</td> </tr> <tr> <td>101_B</td> <td>4.5 transmission clock cycles</td> </tr> <tr> <td>110_B</td> <td>6.5 transmission clock cycles</td> </tr> <tr> <td>111_B</td> <td>8.5 transmission clock cycles</td> </tr> </tbody> </table> | | | CSIHnIDx [2:0] | Idle Time | 000 _B | 0.5 transmission clock a half cycle | 001 _B | 1 transmission clock cycle | 010 _B | 1.5 transmission clock cycles | 011 _B | 2.5 transmission clock cycles | 100 _B | 3.5 transmission clock cycles | 101 _B | 4.5 transmission clock cycles | 110 _B | 6.5 transmission clock cycles | 111 _B | 8.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CSIHnIDx [2:0] | Idle Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 000 _B | 0.5 transmission clock a half cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 001 _B | 1 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 010 _B | 1.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 011 _B | 2.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 100 _B | 3.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 101 _B | 4.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 110 _B | 6.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 111 _B | 8.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| These bits are only available in master mode. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 to 8 | CSIHnHDx[3:0] | Specifies the hold time for chip select signal x in transmission clock cycles. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>CSIHnHDx [3:0]</th> <th>Hold Time with CSIHnCTL1.CSIHnSIT = 0</th> <th>Hold Time with CSIHnCTL1.CSIHnSIT = 1</th> </tr> </thead> <tbody> <tr> <td>0000_B</td> <td>0.5 transmission clock a half cycle</td> <td>1 transmission clock cycle</td> </tr> <tr> <td>0001_B</td> <td>1.0 transmission clock cycle</td> <td>1.5 transmission clock cycle</td> </tr> <tr> <td>0010_B</td> <td>1.5 transmission clock cycles</td> <td>2 transmission clock cycle</td> </tr> <tr> <td>0011_B</td> <td>2.5 transmission clock cycles</td> <td>3.0 transmission clock cycle</td> </tr> <tr> <td>0100_B</td> <td>3.5 transmission clock cycles</td> <td>4.0 transmission clock cycle</td> </tr> <tr> <td>0101_B</td> <td>4.5 transmission clock cycles</td> <td>5.0 transmission clock cycle</td> </tr> <tr> <td>0110_B</td> <td>6.5 transmission clock cycles</td> <td>7.0 transmission clock cycle</td> </tr> <tr> <td>0111_B</td> <td>8.5 transmission clock cycles</td> <td>9.0 transmission clock cycle</td> </tr> <tr> <td>1000_B</td> <td>9.5 transmission clock cycles</td> <td>10.0 transmission clock cycle</td> </tr> <tr> <td>1001_B</td> <td>10.5 transmission clock cycles</td> <td>11.0 transmission clock cycle</td> </tr> <tr> <td>1010_B</td> <td>11.5 transmission clock cycles</td> <td>12.0 transmission clock cycle</td> </tr> <tr> <td>1011_B</td> <td>12.5 transmission clock cycles</td> <td>13.0 transmission clock cycle</td> </tr> <tr> <td>1100_B</td> <td>14.5 transmission clock cycles</td> <td>15.0 transmission clock cycle</td> </tr> <tr> <td>1101_B</td> <td>16.5 transmission clock cycles</td> <td>17.0 transmission clock cycle</td> </tr> <tr> <td>1110_B</td> <td>18.5 transmission clock cycles</td> <td>19.0 transmission clock cycle</td> </tr> <tr> <td>1111_B</td> <td>20.5 transmission clock cycles</td> <td>21 transmission clock cycle</td> </tr> </tbody> </table> | | | CSIHnHDx [3:0] | Hold Time with CSIHnCTL1.CSIHnSIT = 0 | Hold Time with CSIHnCTL1.CSIHnSIT = 1 | 0000 _B | 0.5 transmission clock a half cycle | 1 transmission clock cycle | 0001 _B | 1.0 transmission clock cycle | 1.5 transmission clock cycle | 0010 _B | 1.5 transmission clock cycles | 2 transmission clock cycle | 0011 _B | 2.5 transmission clock cycles | 3.0 transmission clock cycle | 0100 _B | 3.5 transmission clock cycles | 4.0 transmission clock cycle | 0101 _B | 4.5 transmission clock cycles | 5.0 transmission clock cycle | 0110 _B | 6.5 transmission clock cycles | 7.0 transmission clock cycle | 0111 _B | 8.5 transmission clock cycles | 9.0 transmission clock cycle | 1000 _B | 9.5 transmission clock cycles | 10.0 transmission clock cycle | 1001 _B | 10.5 transmission clock cycles | 11.0 transmission clock cycle | 1010 _B | 11.5 transmission clock cycles | 12.0 transmission clock cycle | 1011 _B | 12.5 transmission clock cycles | 13.0 transmission clock cycle | 1100 _B | 14.5 transmission clock cycles | 15.0 transmission clock cycle | 1101 _B | 16.5 transmission clock cycles | 17.0 transmission clock cycle | 1110 _B | 18.5 transmission clock cycles | 19.0 transmission clock cycle | 1111 _B | 20.5 transmission clock cycles | 21 transmission clock cycle |
| CSIHnHDx [3:0] | Hold Time with CSIHnCTL1.CSIHnSIT = 0 | Hold Time with CSIHnCTL1.CSIHnSIT = 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | 0.5 transmission clock a half cycle | 1 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | 1.0 transmission clock cycle | 1.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | 1.5 transmission clock cycles | 2 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 _B | 2.5 transmission clock cycles | 3.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 _B | 3.5 transmission clock cycles | 4.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 _B | 4.5 transmission clock cycles | 5.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 _B | 6.5 transmission clock cycles | 7.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 _B | 8.5 transmission clock cycles | 9.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 _B | 9.5 transmission clock cycles | 10.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 _B | 10.5 transmission clock cycles | 11.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 _B | 11.5 transmission clock cycles | 12.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 _B | 12.5 transmission clock cycles | 13.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 _B | 14.5 transmission clock cycles | 15.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 _B | 16.5 transmission clock cycles | 17.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 _B | 18.5 transmission clock cycles | 19.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | 20.5 transmission clock cycles | 21 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| These bits are only available in master mode. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 14.27 CSIHnCFGx Register Contents (4/4)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|---|--|----------------|---|---|-------------------|-------------------------------------|------------------------------|-------------------|-------------------------------------|------------------------------|-------------------|------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|-------------------------------|------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|--------------------------------|-------------------------------|-------------------|-------------------------------|-------------------------------|
| 7 to 4 | CSIHnINx[3:0] | Specifies the inter-data time for chip select signal x in transmission clock cycles. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>CSIHnINx [3:0]</th> <th>Inter-data time with CSIHnCTL1.CSIHnSIT = 0</th> <th>Inter-data time with CSIHnCTL1.CSIHnSIT = 1</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>0.0 transmission clock a half cycle</td><td>0.5 transmission clock cycle</td></tr> <tr><td>0001_B</td><td>0.5 transmission clock a half cycle</td><td>1.0 transmission clock cycle</td></tr> <tr><td>0010_B</td><td>1.0 transmission clock cycle</td><td>1.5 transmission clock cycle</td></tr> <tr><td>0011_B</td><td>2.0 transmission clock cycles</td><td>2.5 transmission clock cycle</td></tr> <tr><td>0100_B</td><td>3.0 transmission clock cycles</td><td>3.5 transmission clock cycle</td></tr> <tr><td>0101_B</td><td>4.0 transmission clock cycles</td><td>4.5 transmission clock cycle</td></tr> <tr><td>0110_B</td><td>6.0 transmission clock cycles</td><td>6.5 transmission clock cycle</td></tr> <tr><td>0111_B</td><td>8.0 transmission clock cycles</td><td>8.5 transmission clock cycle</td></tr> <tr><td>1000_B</td><td>9.0 transmission clock cycles</td><td>9.5 transmission clock cycle</td></tr> <tr><td>1001_B</td><td>10.0 transmission clock cycles</td><td>10.5 transmission clock cycle</td></tr> <tr><td>1010_B</td><td>11.0 transmission clock cycles</td><td>11.5 transmission clock cycle</td></tr> <tr><td>1011_B</td><td>12.0 transmission clock cycles</td><td>12.5 transmission clock cycle</td></tr> <tr><td>1100_B</td><td>14.0 transmission clock cycles</td><td>14.5 transmission clock cycle</td></tr> <tr><td>1101_B</td><td>16.0 transmission clock cycles</td><td>16.5 transmission clock cycle</td></tr> <tr><td>1110_B</td><td>18.0 transmission clock cycles</td><td>18.5 transmission clock cycle</td></tr> <tr><td>1111_B</td><td>20.0 transmission clock cycle</td><td>20.5 transmission clock cycle</td></tr> </tbody> </table> | CSIHnINx [3:0] | Inter-data time with CSIHnCTL1.CSIHnSIT = 0 | Inter-data time with CSIHnCTL1.CSIHnSIT = 1 | 0000 _B | 0.0 transmission clock a half cycle | 0.5 transmission clock cycle | 0001 _B | 0.5 transmission clock a half cycle | 1.0 transmission clock cycle | 0010 _B | 1.0 transmission clock cycle | 1.5 transmission clock cycle | 0011 _B | 2.0 transmission clock cycles | 2.5 transmission clock cycle | 0100 _B | 3.0 transmission clock cycles | 3.5 transmission clock cycle | 0101 _B | 4.0 transmission clock cycles | 4.5 transmission clock cycle | 0110 _B | 6.0 transmission clock cycles | 6.5 transmission clock cycle | 0111 _B | 8.0 transmission clock cycles | 8.5 transmission clock cycle | 1000 _B | 9.0 transmission clock cycles | 9.5 transmission clock cycle | 1001 _B | 10.0 transmission clock cycles | 10.5 transmission clock cycle | 1010 _B | 11.0 transmission clock cycles | 11.5 transmission clock cycle | 1011 _B | 12.0 transmission clock cycles | 12.5 transmission clock cycle | 1100 _B | 14.0 transmission clock cycles | 14.5 transmission clock cycle | 1101 _B | 16.0 transmission clock cycles | 16.5 transmission clock cycle | 1110 _B | 18.0 transmission clock cycles | 18.5 transmission clock cycle | 1111 _B | 20.0 transmission clock cycle | 20.5 transmission clock cycle |
| CSIHnINx [3:0] | Inter-data time with CSIHnCTL1.CSIHnSIT = 0 | Inter-data time with CSIHnCTL1.CSIHnSIT = 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | 0.0 transmission clock a half cycle | 0.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | 0.5 transmission clock a half cycle | 1.0 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | 1.0 transmission clock cycle | 1.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 _B | 2.0 transmission clock cycles | 2.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 _B | 3.0 transmission clock cycles | 3.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 _B | 4.0 transmission clock cycles | 4.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 _B | 6.0 transmission clock cycles | 6.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 _B | 8.0 transmission clock cycles | 8.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 _B | 9.0 transmission clock cycles | 9.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 _B | 10.0 transmission clock cycles | 10.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 _B | 11.0 transmission clock cycles | 11.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 _B | 12.0 transmission clock cycles | 12.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 _B | 14.0 transmission clock cycles | 14.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 _B | 16.0 transmission clock cycles | 16.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 _B | 18.0 transmission clock cycles | 18.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | 20.0 transmission clock cycle | 20.5 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

These bits are only available in master mode.

| 3 to 0 | CSIHnSPx[3:0] | Specifies the setup time for chip select signal x in transmission clock cycles. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|-------------------------------------|---|---------------|------------|-------------------|-------------------------------------|-------------------|----------------------------|-------------------|-------------------------------|-------------------|-------------------------------|-------------------|-------------------------------|-------------------|-------------------------------|-------------------|-------------------------------|-------------------|-------------------------------|-------------------|-------------------------------|-------------------|--------------------------------|-------------------|--------------------------------|-------------------|--------------------------------|-------------------|--------------------------------|-------------------|--------------------------------|-------------------|--------------------------------|-------------------|--------------------------------|
| | | <table border="1"> <thead> <tr> <th>CSIHnSPx[3:0]</th> <th>Setup time</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>0.5 transmission clock a half cycle</td></tr> <tr><td>0001_B</td><td>1 transmission clock cycle</td></tr> <tr><td>0010_B</td><td>1.5 transmission clock cycles</td></tr> <tr><td>0011_B</td><td>2.5 transmission clock cycles</td></tr> <tr><td>0100_B</td><td>3.5 transmission clock cycles</td></tr> <tr><td>0101_B</td><td>4.5 transmission clock cycles</td></tr> <tr><td>0110_B</td><td>6.5 transmission clock cycles</td></tr> <tr><td>0111_B</td><td>8.5 transmission clock cycles</td></tr> <tr><td>1000_B</td><td>9.5 transmission clock cycles</td></tr> <tr><td>1001_B</td><td>10.5 transmission clock cycles</td></tr> <tr><td>1010_B</td><td>11.5 transmission clock cycles</td></tr> <tr><td>1011_B</td><td>12.5 transmission clock cycles</td></tr> <tr><td>1100_B</td><td>14.5 transmission clock cycles</td></tr> <tr><td>1101_B</td><td>16.5 transmission clock cycles</td></tr> <tr><td>1110_B</td><td>18.5 transmission clock cycles</td></tr> <tr><td>1111_B</td><td>20.5 transmission clock cycles</td></tr> </tbody> </table> | CSIHnSPx[3:0] | Setup time | 0000 _B | 0.5 transmission clock a half cycle | 0001 _B | 1 transmission clock cycle | 0010 _B | 1.5 transmission clock cycles | 0011 _B | 2.5 transmission clock cycles | 0100 _B | 3.5 transmission clock cycles | 0101 _B | 4.5 transmission clock cycles | 0110 _B | 6.5 transmission clock cycles | 0111 _B | 8.5 transmission clock cycles | 1000 _B | 9.5 transmission clock cycles | 1001 _B | 10.5 transmission clock cycles | 1010 _B | 11.5 transmission clock cycles | 1011 _B | 12.5 transmission clock cycles | 1100 _B | 14.5 transmission clock cycles | 1101 _B | 16.5 transmission clock cycles | 1110 _B | 18.5 transmission clock cycles | 1111 _B | 20.5 transmission clock cycles |
| CSIHnSPx[3:0] | Setup time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | 0.5 transmission clock a half cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | 1 transmission clock cycle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | 1.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 _B | 2.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 _B | 3.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 _B | 4.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 _B | 6.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 _B | 8.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 _B | 9.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 _B | 10.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 _B | 11.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 _B | 12.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 _B | 14.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 _B | 16.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 _B | 18.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | 20.5 transmission clock cycles | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

These bits are only available in master mode.

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.12 CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access

This register stores transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1008_H

Value after reset: X0XX XXXX_H

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|--------------|--------------|-----|-----|-----|-----|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CSIHn CIRE | CSIHn EOJ | CSIHn EDL | — | — | — | — | — | CSIHn CS7 | CSIHn CS6 | CSIHn CS5 | CSIHn CS4 | CSIHn CS3 | CSIHn CS2 | CSIHn CS1 | CSIHn CS0 |
| Value after reset | — | — | — | 0 | 0 | 0 | 0 | 0 | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CSIHnTX[15:0] | | | | | | | | | | | | | | | |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 14.28 CSIHnTX0W Register Contents (1/2)

| Bit position | Bit name | Function |
|--------------|-----------|--|
| 31 | CSIHnCIRE | <p>Enables the communication interrupt request INTCSIHnIC in dual buffer or transmit-only buffer mode, or the job completion interrupt request INTCSIHnJC in FIFO mode.</p> <p>0: No interrupt is requested. 1: An interrupt is requested. Generates interrupt INTCSIHnIC or INTCSIHnJC after transmission. For details, see Section 14.4.3, INTCSIHnIC (Communication Status Interrupt) and Section 14.4.6, INTCSIHnJC (Job Completion Interrupt).</p> <p>CAUTION</p> <p>This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1).</p> |
| 30 | CSIHnEOJ | <p>Specifies the end of a job.</p> <p>0: Indicates that it is not end-of-job data. The job continues. 1: Indicates end-of-job data.</p> <p>CAUTION</p> <p>This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). This bit must be set to 0 in slave mode.</p> |
| 29 | CSIHnEDL | <p>Specifies whether the associated data requires the extended data length (EDL) option.</p> <p>0: Normal operation 1: Enables the extended data length. The associated data is transmitted as a 16-bit packet. No inter-data time or idle time will be inserted after the data is transmitted. If CSIHnCTL1.CSIHnEDLE = 1 and CSIHnTX0W.CSIHnEDL = 1, the same CS must be selected for subsequent data. If CS is modified for the subsequent data, the correct operation is not guaranteed.</p> <p>CAUTION</p> <p>This bit is only available if CSIHnCTL1.CSIHnEDLE = 1.</p> |
| 28 to 24 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 14.28 CSIHnTX0W Register Contents (2/2)

| Bit position | Bit name | Function |
|--------------|---------------|---|
| 23 to 16 | CSIHnCS[7:0] | <p>Activates one or more chip select signals.</p> <p>0: Activates chip select signal x for the associated transmission. 1: Deactivates chip select signal x for the associated transmission. Setting CSIHnTX0W.CSIHnCS[7:0] = FF_H is prohibited.</p> <p>CAUTION</p> <p>If several chip select signals are enabled for broadcasting, the configuration of one with CSIHnCFGx.CSIHnRCBx = 0 (dominant) is used. In this case, all dominant chip select signals must be set to precisely the same value. In slave mode, set the CSIHnCS[7:0] bit to FE_H.</p> |
| 15 to 0 | CSIHnTX[15:0] | Stores the transmission data. |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.13 CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

The settings specified by the upper 16 bits of CSIHnTX0W are applied to the transmission. Set transmit data to CSIHnTX0W before using this register because the value of CSIHnTX0W is undefined after the reset.

Access: This register can be read/written in 16-bit units.

Address: <CSIHn_base> + 100C_H

Value after reset: Undefined

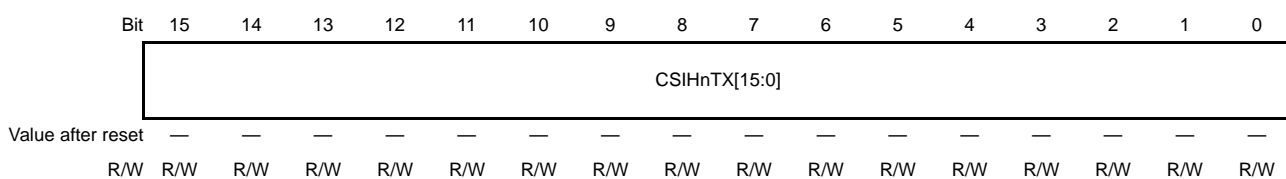


Table 14.29 CSIHnTX0H Register Contents

| Bit position | Bit name | Function |
|--------------|---------------|-------------------------------|
| 15 to 0 | CSIHnTX[15:0] | Stores the transmission data. |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.14 CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access

This register stores the received data.

Access: This register can only be read in 32-bit units.

Address: <CSIHn_base> + 1010_H

Value after reset: 0XXX XXXX_H

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|----|----|----|----|----|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | CSIHn RPE | CSIHn TDCE | CSIHn CS7 | CSIHn CS6 | CSIHn CS5 | CSIHn CS4 | CSIHn CS3 | CSIHn CS2 | CSIHn CS1 | CSIHn CS0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CSIHnRX[15:0] | | | | | | | | | | | | | | | |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 14.30 CSIHnRX0W Register Contents

| Bit position | Bit name | Function |
|--------------|--------------------------|--|
| 31 to 26 | Reserved | When read, the value after reset is read. |
| 25 | CSIHnRPE | Indicates whether a reception data parity error was detected. 0: No parity error was detected on the associated reception data. 1: A parity error was detected on the associated reception data. |
| 24 | CSIHnTDCE | Indicates whether a transmission data consistency check error was detected. 0: No consistency check error was detected on the associated transmission data. 1: A consistency check error was detected on the associated transmission data. |
| 23 to 16 | CSIHnCSx (x = 7 to 0) | Indicates which chip select signal was activated. 0: Chip select signal x is active for the associated reception. 1: Chip select signal x is inactive for the associated reception. |
| 15 to 0 | CSIHnRX[15:0] | Stores the received data. |

NOTE

This register stores the received data when an INTCSIHnIR interrupt is generated. Read the received data stored in this register before generation of an INTCSIHnIR interrupt. Otherwise the data is rewritten with the next received data.

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers**.

14.3.15 CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access

This register stores the received data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

Access: This register can be read only in 16-bit units.

Address: <CSIHn_base> + 1014_H

Value after reset: Undefined

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | CSIHnRX[15:0] | | | | | | | | | | | | | | | |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 14.31 CSIHnRX0H Register Contents

| Bit position | Bit name | Function |
|--------------|---------------|---------------------------|
| 15 to 0 | CSIHnRX[15:0] | Stores the received data. |

NOTE

This register stores the received data when an INTCSIHnIR interrupt is generated. Read the received data stored in this register before generation of an INTCSIHnIR interrupt. Otherwise the data is rewritten with the next received data.

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers.**

14.3.16 CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3)

This register sets the transfer clock frequency for each chip select signal.

With CSIHnCFG0-7.CSIHnBRSSx[1:0] bits, one of the four types of transfer clock frequency settings can be selected for each chip select signal. For details of transfer clock frequency settings, see **Section 14.5.5, Transmission Clock Selection**.

Access: This register can be read/written in 16-bit units.

Address: CSIHnBRS0: <CSIHn_base> + 1068_H
 CSIHnBRS1: <CSIHn_base> + 106C_H
 CSIHnBRS2: <CSIHn_base> + 1070_H
 CSIHnBRS3: <CSIHn_base> + 1074_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | CSIHnBRS[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 14.32 CSIHnBRSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 to 0 | CSIHnBRS [11:0] | 0: BRG stopped 1: PCLK / (2 ^α × 1 × 2) 2: PCLK / (2 ^α × 2 × 2) 3: PCLK / (2 ^α × 3 × 2) 4: PCLK / (2 ^α × 4 × 2) . . . 4095: PCLK / (2 ^α × 4095 × 2) α is the value of CSIHnCTL2.CSIHnPRS[2:0]. |

CAUTION

When setting this register, refer to **Table 14.34, Notes on Setting Registers**.

14.3.17 SELCSIHDMA — CSIH DMA Select Register

This register controls the generation condition for the interrupt and the DMA/DTS trigger.

Access: This register can be read/written in 8-bit units.

Address: <CSIH0_base> + E000_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|-----------|-----------|-----------|-----------|
| | — | — | — | — | SELCSI1DR | SELCSI1DC | SELCSI0DR | SELCSI0DC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 14.33 SELCSIHDMA Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | SELCSI1DR | Controls the generation condition for the INTCSIH1IR0S interrupt and the DMA/DTS trigger. 0: INTCSIH1IR0S interrupt and DMA/DTS trigger can be generated regardless the CS0 setting. 1: INTCSIH1IR0S interrupt and DMA/DTS trigger can be generated only when CS0 is active. |
| 2 | SELCSI1DC | Controls the generation condition for the INTCSIH1IC0S interrupt and the DMA/DTS trigger. 0: INTCSIH1IC0S interrupt and DMA/DTS trigger can be generated regardless the CS0 setting. 1: INTCSIH1IC0S interrupt and DMA/DTS trigger can be generated only when CS0 is active. |
| 1 | SELCSI0DR | Controls the generation condition for the INTCSIH0IR0S interrupt and the DMA/DTS trigger. 0: INTCSIH0IR0S interrupt and DMA/DTS trigger can be generated regardless the CS0 setting. 1: INTCSIH0IR0S interrupt and DMA/DTS trigger can be generated only when CS0 is active. |
| 0 | SELCSI0DC | Controls the generation condition for the INTCSIH0IC0S interrupt and the DMA/DTS trigger. 0: INTCSIH0IC0S interrupt and DMA/DTS trigger can be generated regardless the CS0 setting. 1: INTCSIH0IC0S interrupt and DMA/DTS trigger can be generated only when CS0 is active. |

Figure 14.2 shows a functional overview of the CSIH_n (n = 0, 1) that is connected to the DMA/DTS and the interrupt controller.

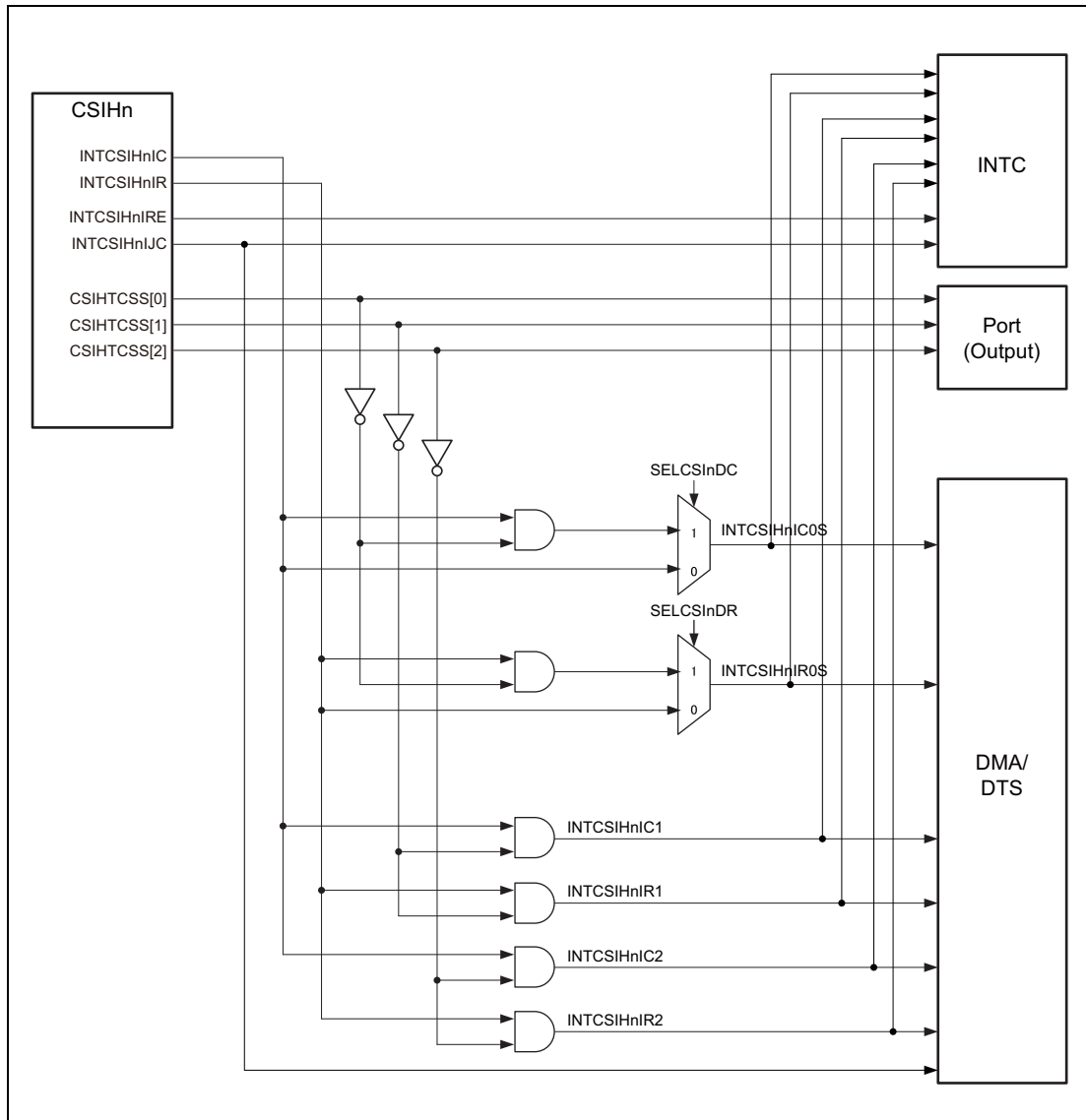


Figure 14.2 Block Diagram Showing the Connection of the Interrupt Controller, DMA/DTC, and CSIH_n (n = 0, 1)

14.3.18 List of Cautions

Table 14.34 Notes on Setting Registers (1/3)

| Register Name | Bit Name | Content |
|---------------|--|--|
| CSIHnCTL0 | CSIHnPWR | If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, the communication must be restarted. |
| CSIHnCTL0 | CSIHnTXE CSIHnRXE | Do not modify any of these bits while CSIHnCTL0.CSIHnPWR = 0. (These bits can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Do not modify these bits while CSIHnSTR0.CSIHnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted. |
| CSIHnCTL0 | CSIHnJOB | Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid when CSIHnCTL1.CSIHnJE = 1. Setting this bit is prohibited in slave mode. |
| CSIHnCTL0 | CSIHnMBS | Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. (This bit can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Modification of this bit is only allowed while CSIHnSTR0.CSIHnTSF = 0. Do not modify of "FIFO mode <-> Direct access mode" while CSIHnCTL0.CSIHnPWR = 1. When the CPU-controlled high-priority communication function is enabled, the operation is the same as that in direct access mode regardless of the CSIHnMBS bit setting. |
| CSIHnCTL1 | CSIHnCKR | Modification of this bit is only allowed while CSIHnCTL0.CSIHnPWR = 0. If CS is not used, use this bit instead of the CSIHnCFGx.CSIHnCKPx bit and set CSIHnCFGx.CSIHnCKPx to 0. This bit must be used in slave mode. |
| CSIHnCTL1 | CSIHnSLIT CSIHnCSL[7:0] CSIHnEDLE CSIHnDCS CSIHnCSRI CSIHnHSE | Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. |
| CSIHnCTL1 | CSIHnPHE CSIHnJE CSIHnLBM | Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of this bit is prohibited in slave mode. |
| CSIHnCTL1 | CSIHnSSE | Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting this bit to 1 is prohibited in master mode. |
| CSIHnCTL1 | CSIHnSIT | Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated. |
| CSIHnCTL2 | CSIHnPRS[2:0] | Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of the maximum transfer clock frequency is shown below. <ul style="list-style-type: none"> • Master mode: 10 MHz • Slave mode: 6.66 MHz |
| CSIHnSTR0 | CSIHnSRP[7:0] CSIHnSPF[7:0] CSIHnFLF CSIHnEMF CSIHnTSF | Writing to these bits is prohibited, and only reading is permitted. |
| CSIHnSTR0 | CSIHnTMOE CSIHnOFFE CSIHnDCE CSIHnPE CSIHnOVE | Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0. |
| CSIHnSTCR0 | CSIHnPCT | Setting this bit to 1 during data transfer stops the transfer, and the interface waits for the next transfer. Setting this bit to 1 during waiting time (setup time, hold time, idle time, or inter-data time), insertion of the waiting time is cancelled and the interface waits for the next transfer. |
| CSIHnMCTL0 | CSIHnMMS[1:0] | Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0 and CSIHnCTL0.CSIHnMBS = 0. |

Table 14.34 Notes on Setting Registers (2/3)

| Register Name | Bit Name | Content |
|-------------------------|---|--|
| CSIHnMCTL0 | CSIHnTO[4:0] | Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0. Set these bits to 0 in master mode. Set these bits to 0 in direct access, dual buffer, and transmit-only buffer mode. |
| CSIHnMCTL1 | CSIHnFES[6:0] CSIHnFFS[6:0] | Writing to these bits while communication is ongoing is permitted. |
| CSIHnMCTL2 | CSIHnBTST CSIHnND[7:0] CSIHnSOP[6:0] | Writing these bits is prohibited when CSIHnCTL0.CSIHnPWR = 0. Writing these bits is prohibited when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0. Writing these bits is prohibited when CSIHnSTR0.CSIHnTSF = 1. Writing these bits is prohibited in direct access or FIFO mode. |
| CSIHnMRWP0 | CSIHnRRA[6:0] | Writing to these bits while communication is ongoing is permitted. Writing these bits is prohibited in direct access or FIFO mode. When writing is required in transmit-only buffer mode, set "0000 _H " to these bits. |
| CSIHnMRWP0 | CSIHnTRWA[6:0] | Writing to these bits while communication is ongoing is permitted. Writing these bits is prohibited in direct access or FIFO mode. |
| CSIHnCFGx x = 0 to 7 | CSIHnBRSSx[1:0] CSIHnRCBx CSIHnIDLx CSIHnIDx[2:0] CSIHnHDx[3:0] CSIHnINx[3:0] CSIHnSPx[3:0] | Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0. These bits must be set to 0 in slave mode. |
| CSIHnCFGx x = 0 to 7 | CSIHnPSx[1:0] CSIHnDLSx[3:0] CSIHnDIRx CSIHnDAPx | Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0. In slave mode, the CSIHnCFG0 setting is used for the configuration. Therefore, all the bits in CSIHnCFG1 to CSIHnCFG7 must be set to 0. |
| CSIHnCFGx x = 0 to 7 | CSIHnCKPx | Modification of this bit is only allowed while CSIHnCTL0.CSIHnPWR = 0. Set this bit to 0 as CSIHnCTL1.CSIHnCKR must be used in slave mode. If CS is not used, use the CSIHnCTL1.CSIHnCKR bit instead of this bit, and clear this bit to 0. |
| CSIHnTX0W | CSIHnEOJ CSIHnCIRE | These bits are only valid when CSIHnCTL1.CSIHnJE = 1. While CSIHnCTL1.CSIHnJE = 0, the value of this bit is ignored even if 1 is read. Set these bits to 0 in slave mode. |
| CSIHnTX0W | CSIHnEDL | This bit is only valid when CSIHnCTL1.CSIHnEDLE = 1. While CSIHnCTL1.CSIHnEDLE = 0, the value of this bit is ignored even if 1 is read. |
| CSIHnTX0W | CSIHnCS[7:0] | In master mode, setting these bits to "FF _H " is prohibited. In slave mode, setting these bits to "FE _H " is prohibited. |
| CSIHnTX0W CSIHnTX0H | | Reading these bits while communication is ongoing is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0 in FIFO mode, reading these bits is prohibited. While CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0, writing to these bits are prohibited in direct access mode. |
| CSIHnRX0W | | These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 1, writing these bits is prohibited and only reading these bits is permitted in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is permitted except for FIFO mode (transmit-only buffer mode, dual buffer mode, and direct access mode). While CSIHnCTL0.CSIHnPWR = 1, writing these bits is prohibited and only reading these bits is permitted except for FIFO mode (transmit-only buffer mode, dual buffer mode, and direct access mode). |

Table 14.34 Notes on Setting Registers (3/3)

| Register Name | Bit Name | Content |
|-------------------------|----------|--|
| CSIHnRX0H | | <p>These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0.</p> <p>While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is prohibited in FIFO mode.</p> <p>While CSIHnCTL0.CSIHnPWR = 1, writing these bits is prohibited and only reading these bits is permitted in FIFO mode.</p> <p>Writing these bits is prohibited and only reading these bits is permitted in spite of the value of CSIHnCTL0.CSIHnPWR except for FIFO mode (transmit-only buffer mode, dual buffer mode, and direct access mode).</p> |
| CSIHnBRSy y = 0 to 3 | | Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. |

14.4 Interrupt Sources

CSIH can generate the following interrupt requests:

- INTCSIHnIC (communication status interrupt)
- INTCSIHnIR (reception status interrupt)
- INTCSIHnIRE (communication error interrupt)
- INTCSIHnIJC (job completion interrupt)

14.4.1 Overview

The error interrupt INTCSIHnIRE is generated when an error is detected. The generation of the other interrupts depends on the memory mode, the job mode, and – in case of the job completion interrupt INTCSIHnIJC – also the operating mode.

The job completion interrupt INTCSIHnIJC is only generated when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). It is not available in slave mode.

The following table gives an overview.

Table 14.35 Interrupt Generation

| Memory mode | Interrupt | Interrupt Source | |
|----------------------|---------------|--|---|
| | | Job Mode Disabled CSIHnCTL1.CSIHnJE = 0 | Job Mode Enabled CSIHnCTL1.CSIHnJE = 1 |
| FIFO | INTCSIHnIC | Tx data empty*1 | Tx data empty*1 Excluding job abort*4 |
| | INTCSIHnIR | Rx data full*2 and CSIHnCTL0.CSIHnRXE = 1 | Rx data full*2 and CSIHnCTL0.CSIHnRXE = 1 |
| | INTCSIHnIRE | Error detected | Error detected |
| | INTCSIHnIJC*3 | Not applicable | CSIHnTX0W.CSIHnCIRE = 1 (When Tx data is not empty), or job abort*4 |
| Transmit-only buffer | INTCSIHnIC | End of communication | CSIHnTX0W.CSIHnCIRE = 1 and (CSIHnCTL0.CSIHnJOBE = 0 or CSIHnTX0W.CSIHnEOJ = 0) |
| | INTCSIHnIR | Data received and CSIHnCTL0.CSIHnRXE = 1 | Data received and CSIHnCTL0.CSIHnRXE = 1 |
| | INTCSIHnIRE | Error detected | Error detected |
| | INTCSIHnIJC*3 | Not applicable | Job abort*4 |
| Dual buffer | INTCSIHnIC | End of communication | CSIHnTX0W.CSIHnCIRE = 1 and (CSIHnCTL0.CSIHnJOBE = 0 or CSIHnTX0W.CSIHnEOJ = 0) |
| | INTCSIHnIR | End of communication and CSIHnCTL0.CSIHnRXE = 1 | Data received and CSIHnCTL0.CSIHnRXE = 1 |
| | INTCSIHnIRE | Error detected | Error detected |
| | INTCSIHnIJC*3 | Not applicable | Job abort*4 |
| Direct access | INTCSIHnIC | One data transfer | One data transfer (Excluding job abort*4) |
| | INTCSIHnIR | Data received and CSIHnCTL0.CSIHnRXE = 1 | Data received and CSIHnCTL0.CSIHnRXE = 1 |
| | INTCSIHnIRE | Error detected | Error detected |
| | INTCSIHnIJC*3 | Not applicable | Job abort*4 |

Note 1. "Tx data empty" refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFES[6:0].

- Note 2. "Rx data full" refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFFS[6:0].
- Note 3. INTCSIHnJC is not available in slave mode.
- Note 4. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1
Same operation as direct access mode is performed during high-priority communication in transmit-only buffer mode.

14.4.2 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half a cycle of the transmission clock, CSIHTSCK. This is not possible in slave mode.

The delay is specified by setting CSIHnCTL1.CSIHnSIT = 1 (the setting of the CSIGNSIT bit is invalid in slave mode).

The following example illustrates the interrupt delay function, assuming a setting of CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled), CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0 (clock and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B (data length 8 bits).

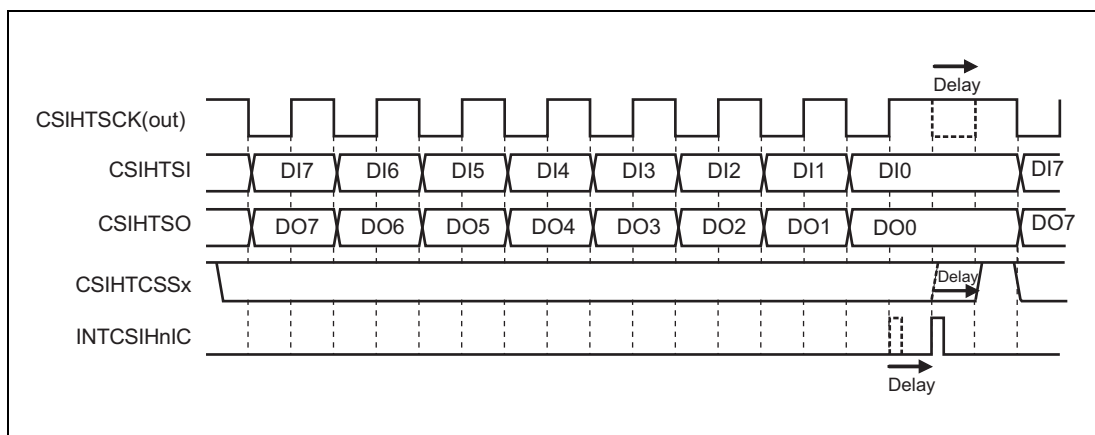


Figure 14.3 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

Setting CSIHnCTL1.CSIHnSIT = 1 adds a half cycle delay to the transmission clock. This also delays the end of the present chip select signal (CSIHTCSSx).

14.4.3 INTCSIHnIC (Communication Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions shown in the following table.

Table 14.36 INTCSIHnIC Interrupt Generation

| Memory Mode | Cause of Interrupt | |
|-----------------------------------|--|--|
| | Job Mode Disabled CSIHnCTL1.CSIHnJE = 0 | Job Mode Enabled CSIHnCTL1.CSIHnJE = 1 |
| FIFO | This interrupt is generated when transmit data FIFO is nearly empty and the application has to add new data to continue transmission. INTCSIHnIC is generated, if the number of data to be transmitted remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMTCL1.CSIHnFES[6:0]. | In the same way as when JE = 0, this interrupt is generated when the number of transmission data CSIHnSTR0.CSIHnSPF[7:0] remaining in the FIFO equals the CSIHnMCTL1.CSIHnFES[6:0] value, but this interrupt is not generated when job is aborted. |
| Transmit-only buffer, dual buffer | Generated at the end of communication. (Specified by the CSIHnMTCL2.CSIHnND[7:0] bit) | Generated when data with CSIHnTX0W.CSIHnCIRE = 1 is transmitted. Note that if data with CSIHnTX0W.CSIHnCIRE = 1 and job abort* ¹ are transmitted, the INTCSIHnJC interrupt is generated instead of INTCSIHnIC. |
| Direct access | Generated after every data transfer. | Generated after every data transfer, except when the communication was aborted. |

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1.
Same operation as direct access mode is performed during high-priority communication in transmit-only buffer mode.

14.4.3.1 INTCSIHnIC in Direct Access Mode

The examples below show the INTCSIHnIC behavior in direct access mode.

The examples assume:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

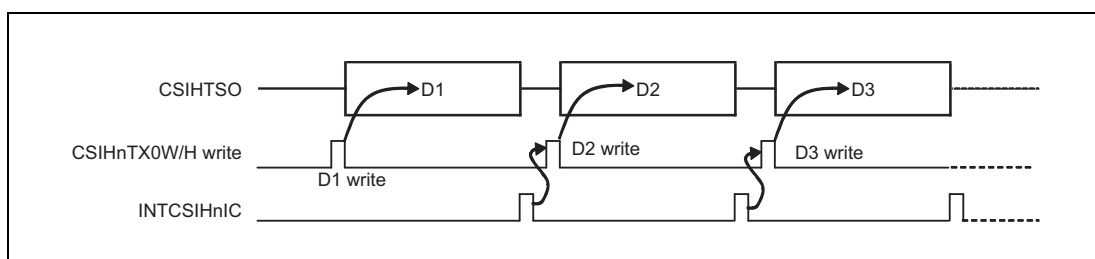


Figure 14.4 Generation of INTCSIHnIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is transmitted with CSIHnTX0W.CSIHnEOJ= 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then INTCSIHnIC is replaced by the job completion interrupt INTCSIHnIC.

INTCSIHnIC can also be set up to occur as soon as the CSIHnTX0W/H register ready to store the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

The effect is illustrated in the figure below.

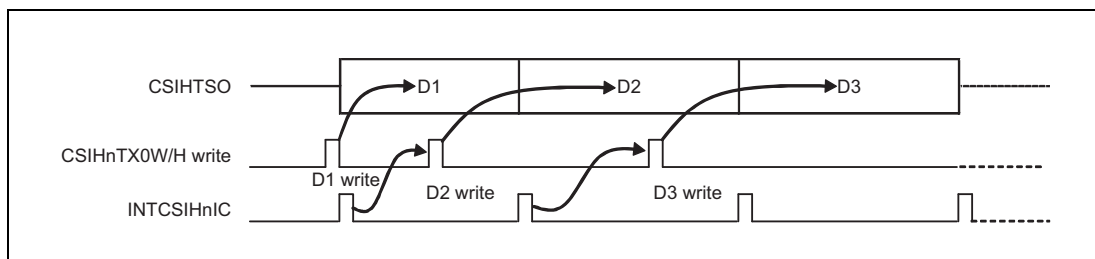


Figure 14.5 Immediate Generation of INTCSIHnIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

NOTE

Same operation as direct access mode is performed during high-priority communication in transmit-only buffer mode.

14.4.3.2 INTCSIHnIC in FIFO Mode

The example below shows the INTCSIHnIC behavior in FIFO mode.

The example assumes:

- Master mode
- FIFO mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase
(CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

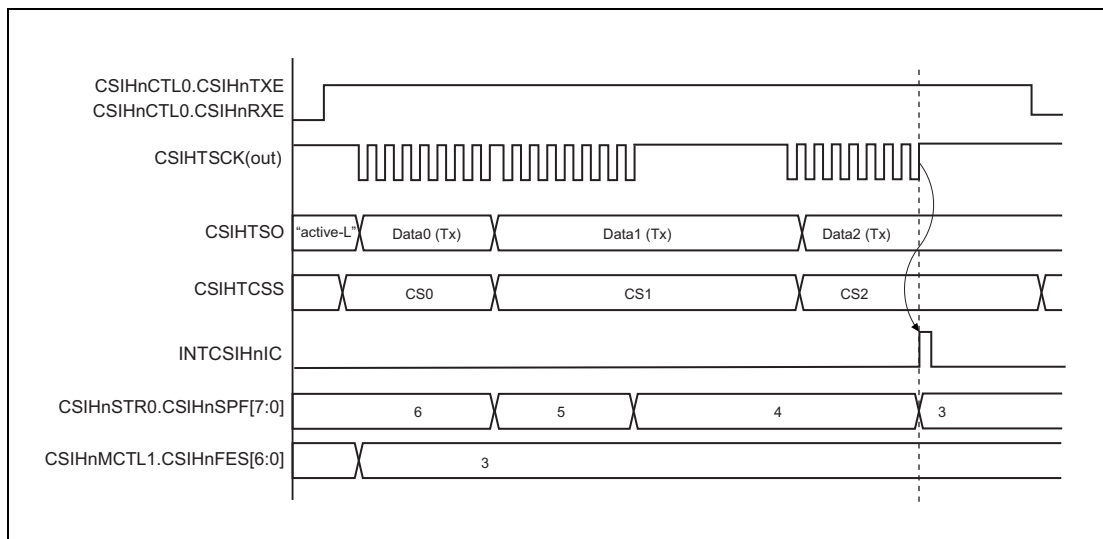


Figure 14.6 Generation of INTCSIHnIC in FIFO Memory Mode

The condition for “FIFO empty” is specified in CSIHnMCTL1.CSIHnFES[6:0]. In the example of the diagram above, the number of untransmitted data in FIFO is set to 3. CSIHnSTR0.CSIHnSPF[7:0] indicates the number of untransmitted data. When both match, the interrupt INTCSIHnIC occurs.

14.4.3.3 INTCSIHnIC in Job Mode

The example below shows the INTCSIHnIC behavior in job mode.

The example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

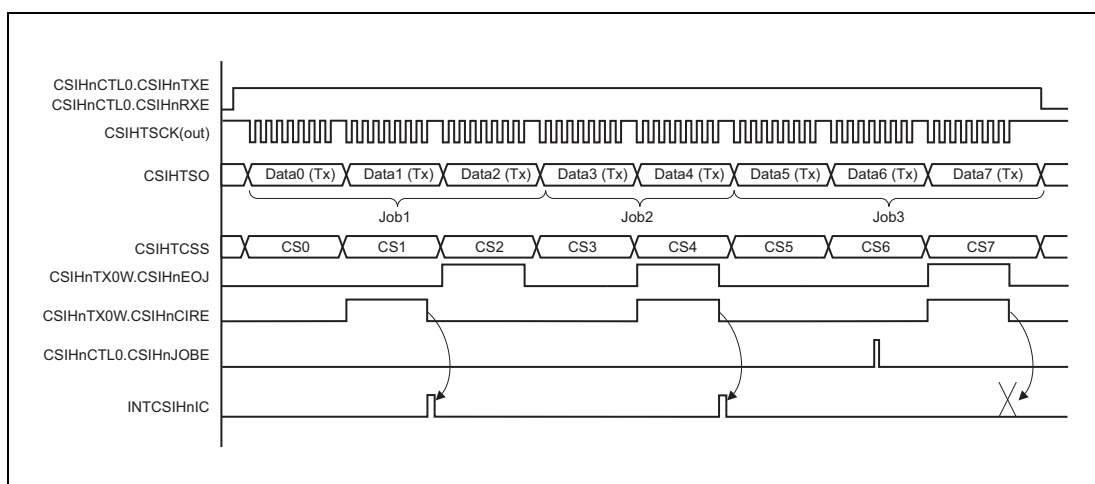


Figure 14.7 Generation of INTCSIHnIC in Job Mode

The rules for generating INTCSIHnIC in job mode are shown in the following table.

Table 14.37 Generation of INTCSIHnIC in Job Mode

| CSIHnTX0W.CSIHnEOJ | CSIHnTX0W.CSIHnCIRE | INTCSIHnIC |
|--------------------|---------------------|--|
| 0 | 0 | Not generated |
| 0 | 1 | Generated |
| 1 | 0 | Not generated |
| 1 | 1 | CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt INTCSIHnJC |

14.4.4 INTCSIHnIR (Receive Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions below.

Table 14.38 INTCSIHnIR Interrupt Generation

| Memory Mode | Cause of Interrupt | |
|-------------------------------------|---|--|
| | CSIHnCTL0.CSIHnRXE = 1 | |
| | Job Mode Disabled CSIHnCTL1.CSIHnJE = 0 | Job Mode Enabled CSIHnCTL1.CSIHnJE = 1 |
| FIFO | This interrupt occurs when the FIFO buffer is almost full with received data while CSIHnCTL0.CSIHnRXE = 1, notifying the application that the FIFO must be emptied. INTCSIHnIR is generated, if the number of received data in the FIFO (CSIHnSTR0.CSIHnSRP[7:0]) equals (128-CSIHnMCTL1.CSIHnFFS[6:0]) | |
| Dual buffer | An interrupt is generated at the end of communication (specified by CSIHnMCTL2.CSIHnND[7:0]) while CSIHnCTL0.CSIHnRXE = 1. | An interrupt is generated after every data transfer. |
| Transmit-only buffer, Direct access | An interrupt is generated after every data transfer. | |

14.4.4.1 INTCSIHnIR in Direct Access Mode

The example below shows the INTCSIHnIR behavior in direct access mode.

The example below assumes:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

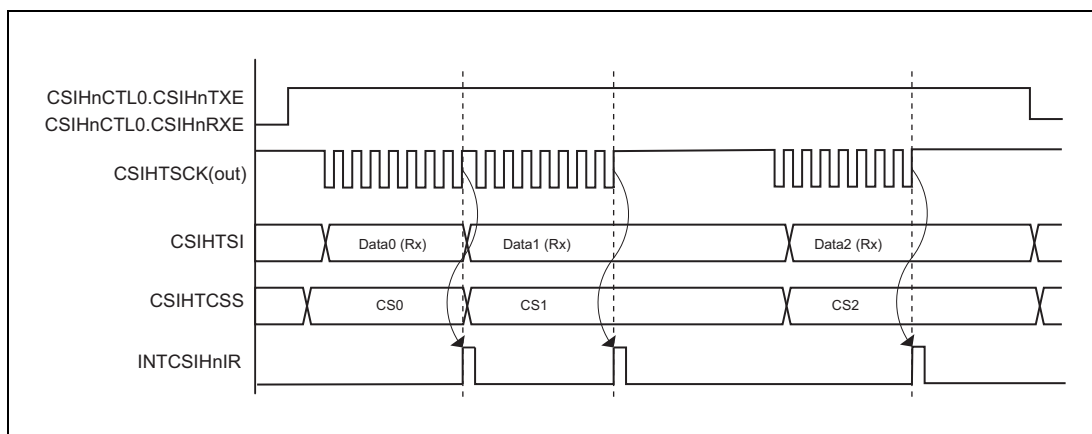


Figure 14.8 Generation of INTCSIHnIR in Direct Access Mode

14.4.4.2 INTCSIHnIR in Dual Buffer Mode

The example below shows the INTCSIHnIR behavior in dual buffer mode.

The example assumes:

- Master mode
- Dual buffer mode
- No interrupt delay ($\text{CSIHnCTL1.CSIHnSIT} = 0$)
- Normal clock and data phase ($\text{CSIHnCFGx.CSIHnCKPx} = 0$, $\text{CSIHnCFGx.CSIHnDAPx} = 0$)
- Data length 8 bits ($\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 1000_{\text{B}}$)

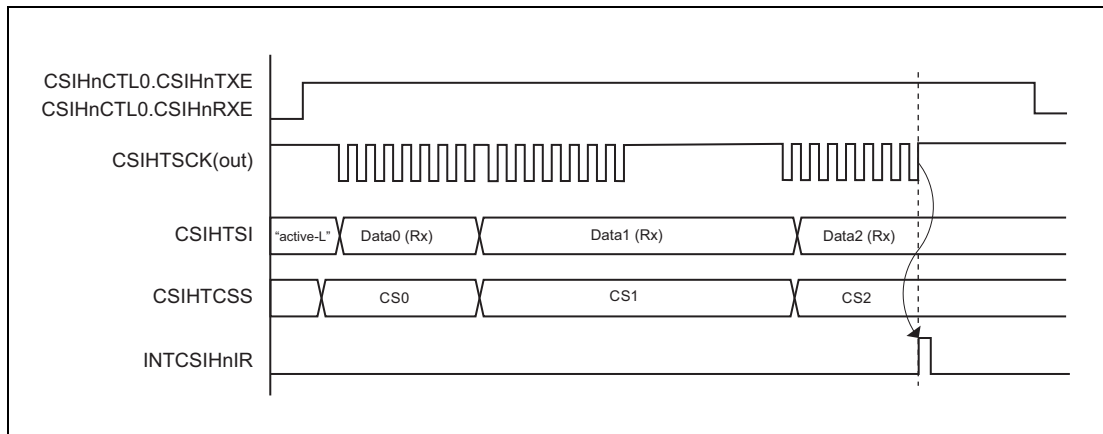


Figure 14.9 Generation of INTCSIHnIR in Dual Buffer Mode

14.4.5 INTCSIHnIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

For details about interrupt generation timing, see **Section 14.5.12, Error Detection**.

Table 14.39 Data Error Types

| Error Type | Communication Status after Error Interrupt | Note |
|------------------------------|--|---|
| FIFO overflow error | Communication continues even if an interrupt is generated. | Data that has not been written to the FIFO buffer and the data overflowed is lost, but communication started before the error is continued. |
| Parity error | Communication continues even if an interrupt is generated. | — |
| Data consistency check error | Communication continues even if an interrupt is generated. | — |
| Time-out error | Communication continues even if an interrupt is generated. | — |
| Overrun error | (Error occurrence condition 1) An interrupt is generated when the CPU reads the CSIHnRX0W/H register while no receive data is present in FIFO mode, but communication continues. | — |
| | (Error occurrence condition 2) When CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled) in slave mode | When CSIHnCTL1.CSIHnHSE = 1 (handshake function enabled) in slave mode, communication stops due to handshake and no overrun error occurs. |
| | [1] An interrupt is generated when reception is completed while the previous receive data remains in the CSIHnRX0W/H register in direct access mode or transmit-only buffer mode, but communication continues. | |
| | [2] An interrupt is generated when reception is completed while the FIFO buffer is filled with received data in FIFO mode, but communication continues. | |

The type of error that caused the generation of INTCSIHnIRE is flagged in register CSIHnSTR0.

Additionally a parity error flag and a data consistency check error flag are attached to the received data in CSIHnRX0W.

For details about the various error types, see **Section 14.5.12, Error Detection**.

14.4.6 INTCSIHnIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs, see **Section 14.5.3.3, Job Concept**. This interrupt is only available in master mode.

Job mode is enabled by setting $CSIHnCTL1.CSIHnJE = 1$. When $CSIHnCTL1.CSIHnJE = 0$, INTCSIHnIJC is not generated.

Depending on the memory mode, this interrupt is generated according to the condition shown in the following table.

Table 14.40 INTCSIHnIJC Interrupt Generation

| Memory Mode | Cause of Interrupt | |
|----------------------|--|---|
| | Job Mode Disabled $CSIHnCTL1.CSIHnJE = 0$ | Job Mode Enabled $CSIHnCTL1.CSIHnJE = 1$ |
| FIFO | Not applicable | <ul style="list-style-type: none"> Indicates that communication stopped at the end of job after job abort*¹ is triggered. When FIFO empty is not detected, INTCSIHnIJC is generated when $CSIHnCIRE = 1$. |
| Transmit-only buffer | | <ul style="list-style-type: none"> Indicates that communication stopped at the end of job after job abort*¹ was triggered. |
| Dual buffer | | |
| Direct access | | |

Note 1. Job abort condition: $CSIHnTX0W.CSIHnEOJ = 1$ and $CSIHnCTL0.CSIHnJOBE = 1$

14.5 Operation

14.5.1 Operating Modes (Master/Slave)

Whether CSIH operates in the master or slave mode determines the source of the serial clock.

14.5.1.1 Master mode

In master mode, the serial transmission clock is generated by the internal built-in baud rate generator (BRG) and supplied to the slave(s) via CSIH_TSCK signal.

Master mode is enabled by setting CSIH_nCTL2.CSIH_nPRS[2:0] to any value but 111_B. In master mode, the BRG frequency can be specified by setting the CSIH_nCTL2.CSIH_nPRS[2:0] bits in combination with the CSIH_nBRSy.CSIH_nBRS[11:0] bits.

(1) Chip select signals

In master mode, one or more chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to select one or more slaves. Only the selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details see **Section 14.5.3, Chip Selection (CS) Features**.

(2) Clock defaults

The default level of CSIH_TSCK depends on the clock phase selection bit: It is high when CSIH_nCTL1.CSIH_nCKR = 0, and is low when CSIH_nCTL1.CSIH_nCKR = 1.

The example below shows communication in master mode for 8-bit data, with CSIH_nCTL1.CSIH_nCKR = 0, CSIH_nCFGx.CSIH_nCKPx = 0, and CSIH_nCFGx.CSIH_nDAPx = 0, and MSB first.

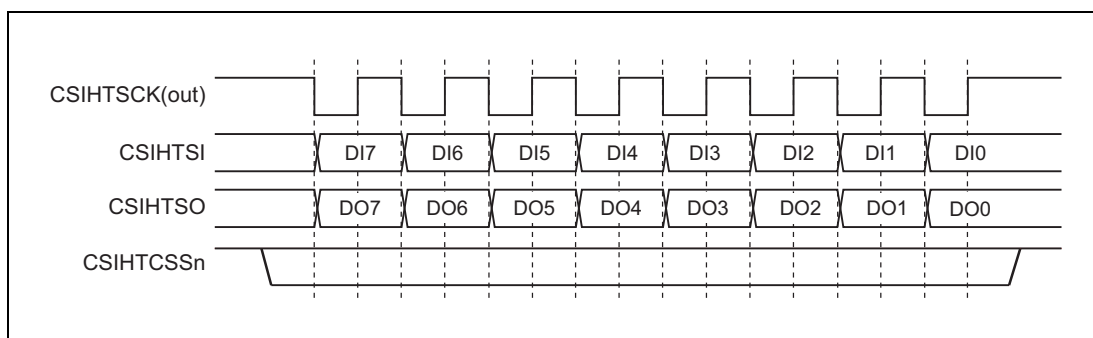


Figure 14.10 Transmission/Reception in Master Mode

14.5.1.2 Slave Mode

In slave mode, another device is the communication master and supplies the transmission clock. Usual transmit operation or receive operation starts immediately after a clock signal is detected.

Slave mode is selected by setting the CSIHnCTL2.CSIHnPRS[2:0] bits to 111_B.

In slave mode, the transmission protocol setting by the CSIHnCFG0 register is enabled (setting of the CSIHnCFG1-CSIHnCFG7 register is disabled).

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx,CSIHnDAPx: clock and data phase

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting bits CSIHnBRSy.CSIHnBRS[11:0] to 000_H. Set the CSIHnBRSy.CSIHnBRS[11:0] bits to a value other than 000_H to use time-out errors.

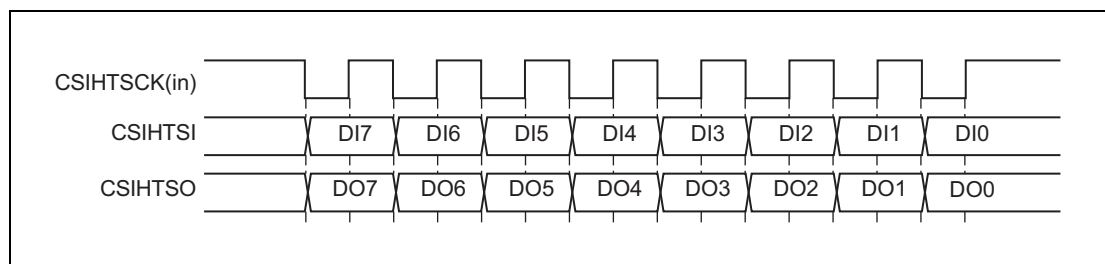


Figure 14.11 Transmission/Reception in Slave Mode

14.5.2 Master/Slave Connections

14.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

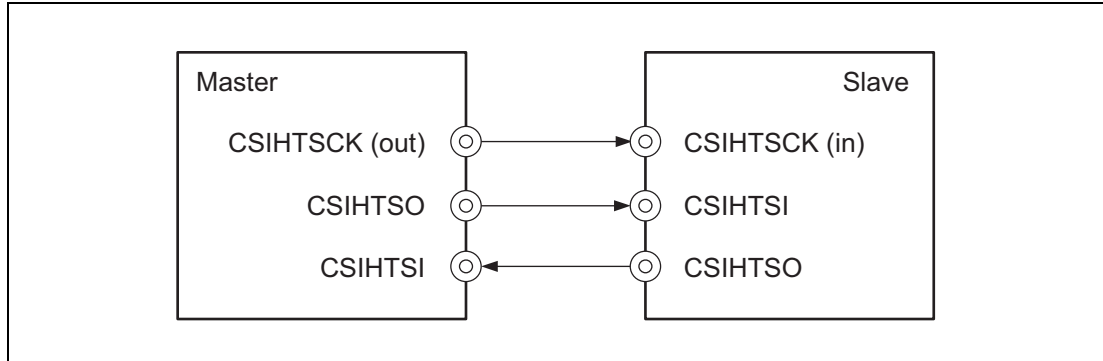


Figure 14.12 Direct Master/Slave Connection

14.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master supplies one chip select (CS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIHTSSI}}$ of the slave.

The $\overline{\text{CSIHTSSI}}$ signal can be enabled or disabled by using the $\text{CSIHnCTL1.CSIHnSSE}$ bit.

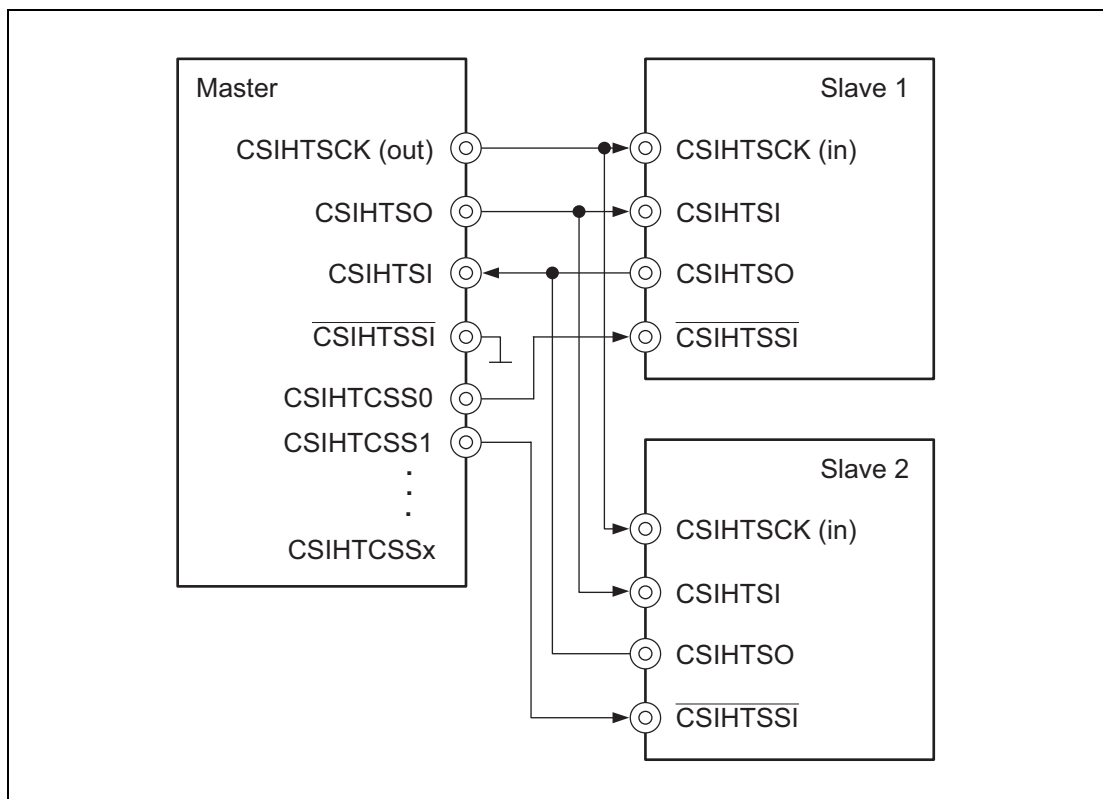


Figure 14.13 One Master to Multiple Slaves Connection

By default, the chip select level is active low. That means, a slave is selected (enabled) as a CSH slave when its $\overline{\text{CSIHTSSI}}$ signal is at a low level. However, to adapt the CS to other devices, the output level of each chip select signal can also be programmed to be active high.

If a slave is not selected, it will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set ($\text{CSIHnCTL0.CSIHnTXE} = 1$), the CSIHTSO output of the slaves that are not selected is disabled and set to input mode in order to avoid interference with the output of the selected slave.

14.5.3 Chip Selection (CS) Features

The chip select signal, CSIHnCSSx can be used by the master to select one or more slaves for communication.

14.5.3.1 Configuration Registers

The parameters for each chip select signal CSIHnCSSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be transmitted or received. (CSIHnCFGx.CSIHnDLSx[3:0])
- Transfer direction: MSB or LSB first. (CSIHnCFGx.CSIHnDIRx)
- Parity usage: Odd, even, 0 parity or none. (CSIHnCFGx.CSIHnPSx[1:0])
- Clock phase and data phase. (CSIHnCFGx.CSIHnCKPx, CSIHnCFGx.CSIHnDAPx)

Additional parameters for each chip select signal that are only available in master mode are:

- Selection of prescaler of baud rate generator individually for each chip select signal (CSIHnCFGx.CSIHnBRSSx[1:0])
- Categorizes chip select signals into “dominant” and “recessive”. The priority applies if two or more chip select signals with different configurations are simultaneously activated for message broadcasting. In this case, the configuration specified for dominant chip select signals is used. (CSIHnCFGx.CSIHnRCBx)

The principle is also called “Recessive Configuration for Broadcasting” (RCB).

CAUTION

When specifying multiple chip select signals as dominant, be sure to configure the same settings for all dominant signals.

- Chip select timing:
 - Setup time T_{setup} : The time from “CS active” to “First data output”. (CSIHnCFGx.CSIHnSPx[3:0])
 - Inter-data time T_{inter} : The time between one data and the next data while the same CS signal is active. (CSIHnCFGx.CSIHnINx[3:0])
 - Hold time T_{hold} : The time from “the end of last data bit” to “CS inactive”. (CSIHnCFGx.CSIHnHDx[3:0])
 - Idle time T_{idle} : Inactive time after terminating a CS signal or after every data transfer to the same CSx. (CSIHnCFGx.CSIHnIDx[2:0])

The CS timings of the setup time, the inter-data time, the hold time, and the idle time are illustrated in the figure below. When the CSIHnCFGx.CSIHnIDLx bit is set to 1, idle time is inserted into each transfer regardless of the CS signal.

Figure 14.14 provides an example of a waveform when the default active low setting is specified for the CSIHnCTL1 and CSIHnCTL2 signals (CSIHnCTL1.CSIHnCSL1 bit = 0, CSIHnCTL1.CSIHnCSL2 bit = 0). The active level can be specified individually for each CS.

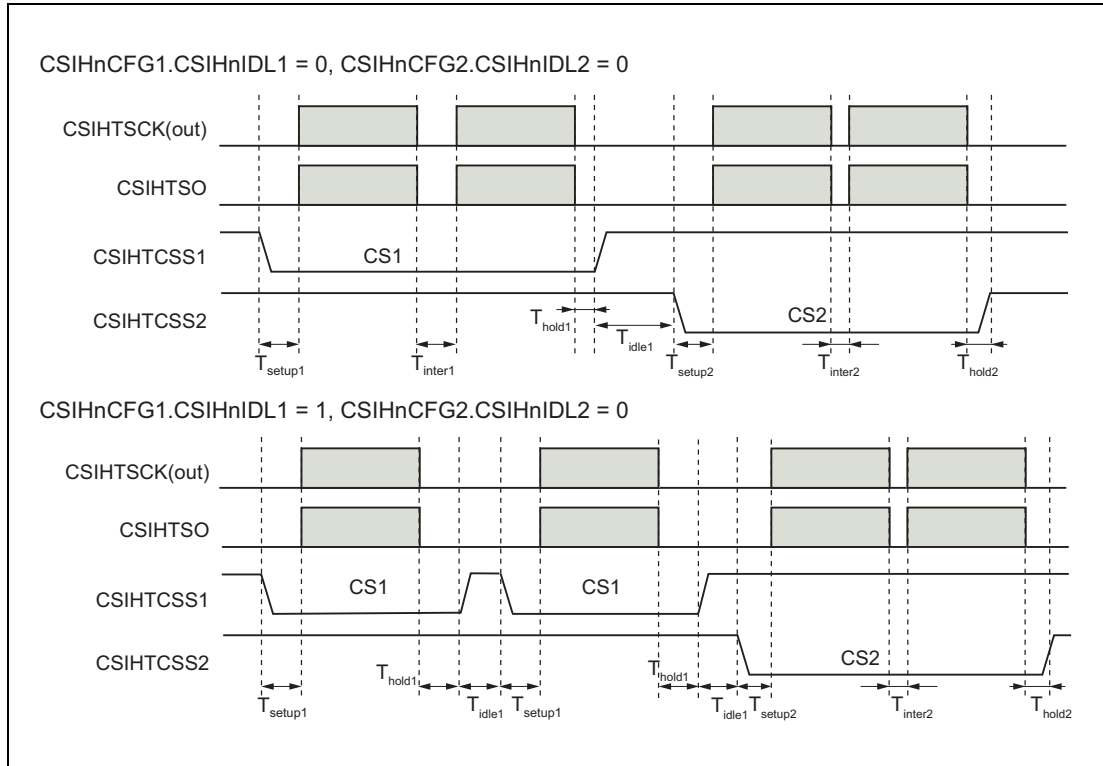


Figure 14.14 Chip Select Timings

Note that each CS signal can have a different value for the setup time, inter-data time, hold time, and idle time.

A particular chip select signal is activated by setting the appropriate bit in the transmission register CSIHnTX0W.CSIHnCSx.

CSIHnRX0W.CSIHnCSx in the reception register indicates the chip select signal associated with the received data.

CAUTION

When high-priority communication function using CPU control is enabled (CSIHnCTL1.CSIHnPHE = 1), idle state is inserted when low-priority communication mode changes to high-priority communication mode and when high-priority communication mode changes to low-priority communication mode regardless of the CSIHnCFGx.CSIHnIDLx bit setting.

14.5.3.2 CS Example

The following figure shows an example of two consecutive data transmissions.

The first communication uses CS0 to communicate with one slave. The second enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority” and the priority of CS1 to “dominant: high priority”.

Therefore, the second communication is performed by using the dominant CS1 setting.

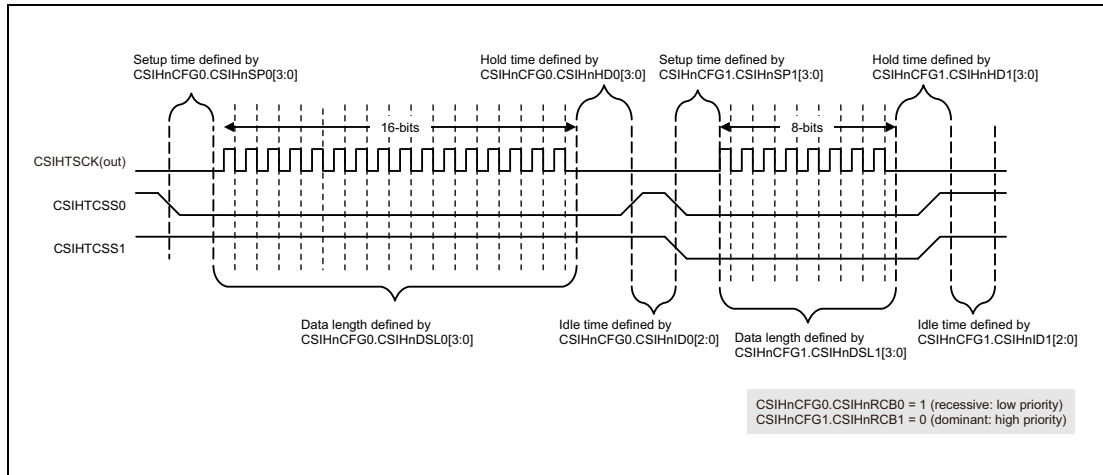


Figure 14.15 Chip Select and RCB Example

14.5.3.3 Job Concept

In CSIH, a job consists of the number of data that are transferred.

Job mode enable

The job mode can only be enabled in master mode. The job mode is enabled and disabled by CSIHnCTL1.CSIHnJE, while the CSIH is disabled by CSIHnCTL0.CSIHnPWR = 0.

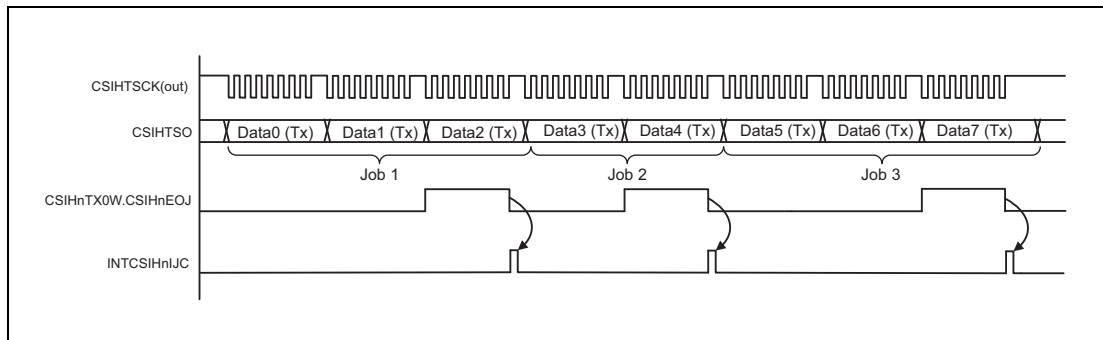


Figure 14.16 Job Examples

A job ends by transmitting data with CSIHnTX0W.CSIHnEOJ = 1.

Communication can be specified to stop when a job is finished. This is done by setting CSIHnCTL0.CSIHnJOB. When CSIHnJOB is set, the communication continues until data is transmitted, for which the CSIHnEOJ bit was set. After this data is transmitted, the communication is stopped and the job completion interrupt INTCSIHnIJC is generated.

14.5.4 Chip Select Timing Details

14.5.4.1 Changing the Clock Phase

The serial clock level specified by $CSIHnCFGx.CSIHnCKPx$ may be changed when communication is disabled. The minimum value of an idle time is one transmission clock ($CSIHTSCK(out)$) cycle.

If the idle time is set to 0.5 transmission clock periods (in $CSIHnCFGx.CSIHnIDx[2:0]$) and two consecutive data are transmitted with different $CSIHnCFGx.CSIHnCKPx$ configuration, the idle time is automatically extended to one $CSIHTSCK(out)$ cycle.

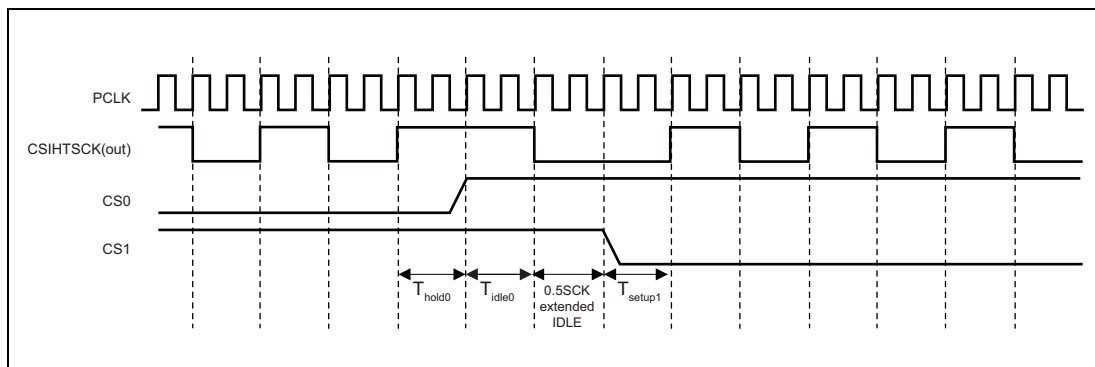


Figure 14.17 Clock Phase Timing with $PCLK/4$, $T_{hold0} = T_{setup1} = 0.5CSIHTSCK$, $T_{idle0} = 0.5CSIHTSCK$, $CSIHnCFG0.CSIHnCKP0 = 0$ (CSIHTCSS0) → $CSIHnCFG1.CSIHnCKP1 = 1$ (CSIHTCSS1)

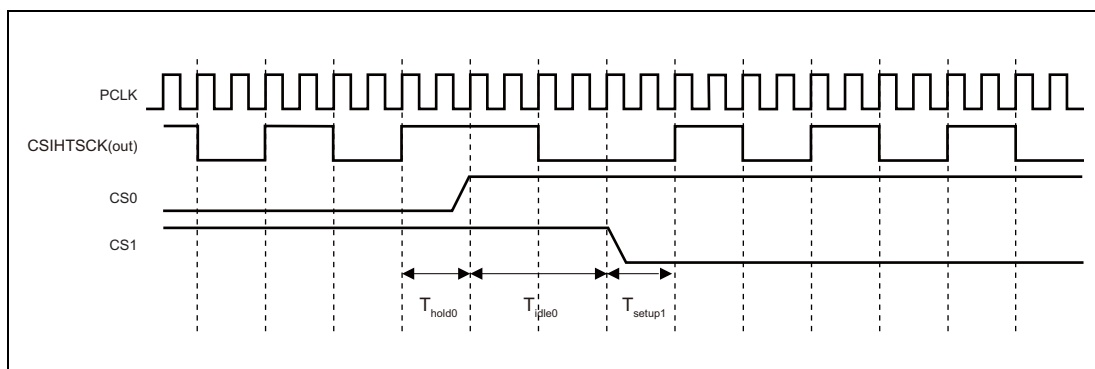


Figure 14.18 Clock Phase Timing with $PCLK/4$, $T_{hold0} = T_{setup1} = 0.5CSIHTSCK$, $T_{idle0} = 1CSIHTSCK$, $CSIHnCFG0.CSIHnCKP0 = 0$ (CSIHTCSS0) → $CSIHnCFG1.CSIHnCKP1 = 1$ (CSIHTCSS1)

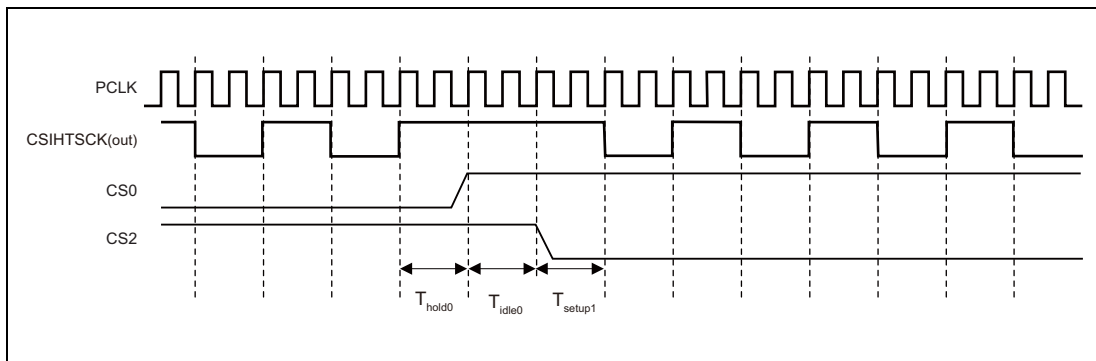


Figure 14.19 Clock Phase Timing with PCLK/4, $T_{hold0} = T_{setup1} = 0.5CSIHTSCK$, $T_{idle0} = 0.5CSIHTSCK$, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG2.CSIHnCKP2 = 0 (CSIHTCSS2)

14.5.4.2 Changing the Data Phase

The CSIHnCFGx.CSIHnDAPx bit defines the phase of the data bits relative to the clock.

The relation between the setting of the CSIHnCFGx.CSIHnDAPx bit and the hold and setup times is as follows:

The hold time period is a period from the last edge of the serial clock (CSIHTSCK) until CSIHnCSS[7:0] becomes inactive level regardless of the CSIHnCFGx.CSIHnDAPx setting.

The setup time period is a period from the time when CSIHnCSS[7:0] becomes active level until transmission data (CSIHTSO) is output.

For this reason, there is a time difference of 0.5 CSIHTSCK cycles until the serial clock (CSIHTSCK) edge is output depending on the CSIHnCFGx.CSIHnDAPx setting.

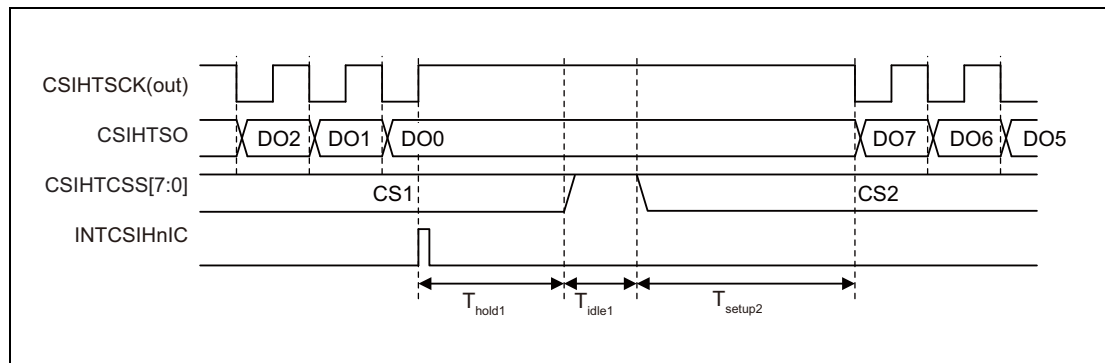


Figure 14.20 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

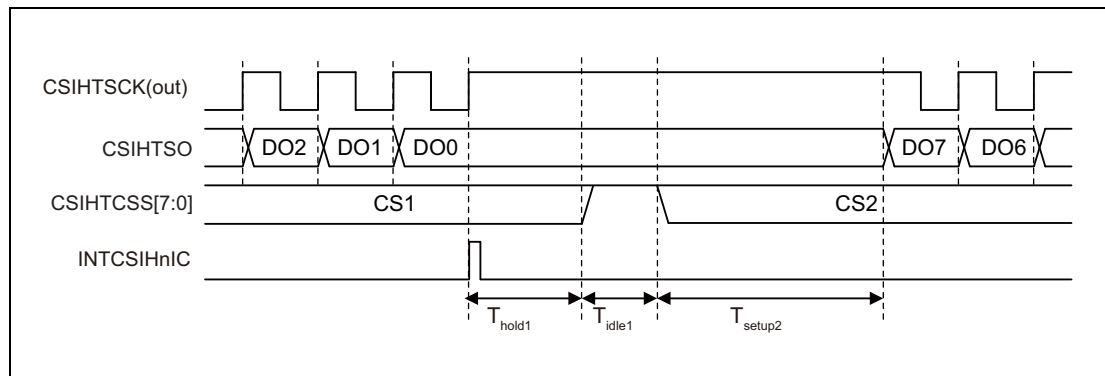


Figure 14.21 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

14.5.5 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the following bits:

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnBRSy.CSIHnBRS[11:0] (y = 0-3)
- CSIHnCFGx.CSIHnBRSSx[1:0] (x = 0 to 7)

The transfer clock frequency of the transmit clock CSIHnTSCk is determined by the CSIHnCTL2.CSIHnPRS[2:0] and CSIHnBRSy.CSIHnBRS [11:0] settings. One among four baud rates, CSIHnBRS3 to CSIHnBRS0, is selected for each chip select signal by the CSIHnCFGx.CSIHnBRSSx[1:0] setting.

The following figure shows a block diagram of the baud rate generator.

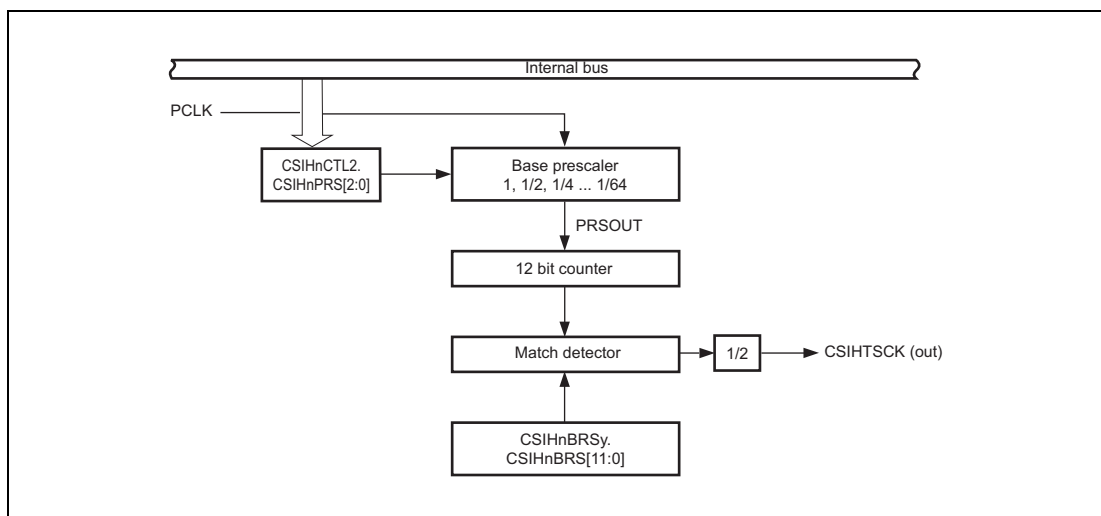


Figure 14.22 Baud Rate Generator Block Diagram

Setting CSIHnBRSy.CSIHnBRS[11:0] to 000_H disables the baud rate generator, and thus all CSIHnTSCk are stopped.

Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated by the following formula.

$$\text{Transfer clock frequency (CSIHnTSCk)} = \text{PCLK} / (\text{PCLK's division ratio}) = \text{PCLK} / (2^\alpha \times k \times 2)$$

where:

$$\alpha = \text{CSIHnCTL2.CSIHnPRS}[2:0] = 0 \text{ to } 6$$

$$k = \text{CSIHnBRS0.CSIHnBRS0}[11:0] = 1 \text{ to } 4095$$

(when CSIHnCFGx.CSIHnBRSSx[1:0] = 0)

$$\text{CSIHnBRS1.CSIHnBRS1}[11:0] = 1 \text{ to } 4095$$

(when CSIHnCFGx.CSIHnBRSSx[1:0] = 1)

$$\text{CSIHnBRS2.CSIHnBRS2}[11:0] = 1 \text{ to } 4095$$

(when CSIHnCFGx.CSIHnBRSSx[1:0] = 2)

$$\text{CSIHnBRS3.CSIHnBRS3}[11:0] = 1 \text{ to } 4095$$

(when CSIHnCFGx.CSIHnBRSSx[1:0] = 3)

Transfer clock frequency limits

When setting the transfer clock frequency, please note the followings:

- Minimum transfer clock frequency = $PCLK / 524160$ (both master mode and slave mode)
- Maximum transfer clock frequency is as follows:
 - Master mode: 10.0 MHz
 - Slave mode: 6.66 MHz

14.5.6 CSIH Buffer Memory

The CSIH has a configurable RAM that can be used for buffered I/O. The size is 128 words. One word is comprised of 32-bit data and 7-bit ECC.

The following configurations are available:

| Mode | CSIHnCTL0. CSIHnMBS | CSIHnMCTL0. CSIHnMMS[1:0] |
|---------------------------|------------------------|------------------------------|
| FIFO mode | 0 | 00 _B |
| Dual buffer mode | | 01 _B |
| Transmit-only buffer mode | | 10 _B |
| Direct access mode | 1 | X |

14.5.6.1 FIFO Mode

In FIFO mode, data can be written to the CSIHnTX0W register without waiting for completion of the transmission, and data can be received without reading the CSIHnRX0W register immediately, provided that the FIFO is not full.

Transmission and reception occur simultaneously - one data is transmitted, one data is received. That means, received data overwrites the transmitted data in the FIFO.

The CSIH automatically updates the respective FIFO memory pointers when data is written to or read from the FIFO memory or transmitted from or received to the FIFO memory.

Table 14.41 FIFO Mode

| Pointer Description | Control Bits*1 | Range |
|--|---------------------------|--|
| Number of untransmitted words | CSIHnSTR0.CSIHnSPF[7:0] | 0 to 128 |
| Number of words received and stored in the FIFO | CSIHnSTR0.CSIHnSRP[7:0] | 0 to 128 |
| Address of transmission data to be written or read | CSIHnMRWP0.CSIHnTRWA[6:0] | 0000 _H to 01FC _H |
| Address of receive data to be read | CSIHnMRWP0.CSIHnRRA[6:0] | 0000 _H to 01FC _H |
| Transmission address | CSIHnMCTL2.CSIHnSOP[6:0] | 0000 _H to 01FC _H |

Note 1. The value is automatically updated each time data is written, read, transmitted, or received.

The CSIH status register contains two FIFO status flags:

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

When this mode is started, bit CSIHnSTCR0.CSIHnPCT must be set. This sets only CSIHnSTR0.CSIHnEMF, but not reset).

All FIFO pointers and FIFO flags except CSIHnSTR0.CSIHnEMF are reset and CSIHnSTR0.CSIHnEMF is set.

14.5.6.2 Dual Buffer Mode

In this mode, the memory is divided into two parts of equal size – this means the lower-order 64 words for transmit data and the higher-order 64 words for received data. In dual buffer mode, the respective buffer pointers indicate the values shown in the following table.

Table 14.42 Dual Buffer Mode

| Pointer Description | Pointer* ¹ | Range |
|---|---------------------------|--|
| Address of data to be written to or read from transmit buffer | CSIHnMRWP0.CSIHnTRWA[6:0] | 0000 _H to 00FC _H |
| Address of data to be read from receive buffer | CSIHnMRWP0.CSIHnRRA[6:0] | 0000 _H to 00FC _H |
| Number of transmission data remaining in transmit buffer | CSIHnMCTL2.CSIHnND[6:0] | 0 to 64 |
| Transmission address | CSIHnMCTL2.CSIHnSOP[6:0] | 0000 _H to 00FC _H |

Note 1. Pointers are automatically incremented each time data is written or read.

14.5.6.3 Transmit-Only Buffer Mode

In this mode, the entire memory is used to save transmission data.

Received data must be read directly from CSIHnRX0W/H.

In transmit-only buffer mode, the respective buffer pointer indicates the values shown in the following table.

Table 14.43 Transmit-Only Buffer Mode

| Pointer Description | Pointer* ¹ | Range |
|---|---------------------------|--|
| Address of data to be written to or read from transmit buffer | CSIHnMRWP0.CSIHnTRWA[6:0] | 0000 _H to 01FC _H |
| Number of transmission data remaining in transmit buffer | CSIHnMCTL2.CSIHnND[6:0] | 0 to 128 |
| Transmission address | CSIHnMCTL2.CSIHnSOP[6:0] | 0000 _H to 01FC _H |

Note 1. Pointers are automatically incremented each time data is written or read.

14.5.6.4 Direct Access Mode

In direct access mode, the CSIH memory is completely bypassed:

- Transmission data provided by the CPU to the transmission register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception register CSIHnRX0W or CSIHnRX0H.

14.5.7 Data Transfer Modes

14.5.7.1 Transmit-Only Mode

Setting $\text{CSIHnCTL0.CSIHnTXE} = 1$ and $\text{CSIHnCTL0.CSIHnRXE} = 0$ puts the CSIH in transmit-only mode. Start of transmission depends on the memory mode:

- In case of FIFO or direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, transmission starts when the $\text{CSIHnMCTL2.CSIHnBTST}$ bit is set.

14.5.7.2 Receive-Only Mode

Setting $\text{CSIHnCTL0.CSIHnTXE} = 0$ and $\text{CSIHnCTL0.CSIHnRXE} = 1$ puts the CSIH in receive-only mode.

In master mode, start of reception depends on the memory mode:

- In case of FIFO or direct access mode, reception starts when dummy data is written in the CSIHnTX0W or CSIHnTX0H register.

In slave mode, reception starts as soon as the CSIHTSCK transmission clock is received from the master. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

- In dual-buffer mode or transmit-only buffer mode, reception starts when the $\text{CSIHnMCTL2.CSIHnBTST}$ bit is set.

14.5.7.3 Transmit/Receive Mode

Setting $\text{CSIHnCTL0.CSIHnTXE} = 1$ and $\text{CSIHnCTL0.CSIHnRXE} = 1$ puts the CSIH in transmit/receive mode.

Start of communication (transmission and reception) depends on the memory mode:

- In case of FIFO or direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, communication starts when the $\text{CSIHnMCTL2.CSIHnBTST}$ bit is set.

14.5.7.4 Summary

The following table summarizes this section. It shows how data transfer is started in the various memory, operating, and transfer modes.

Table 14.44 Start of Data Transfer

| Memory and Operating Mode | | Transfer Mode | |
|-----------------------------------|--------|---|---|
| | | Transmit-only Transmit/Receive | Receive-only |
| FIFO, direct access | Master | Writing to the CSIHnTX0W register or the CSIHnTX0H register | Writing to the CSIHnTX0W register or the CSIHnTX0H register |
| | Slave | Clock supplied from the master | Clock supplied from the master |
| Transmit-only buffer, dual buffer | Master | $\text{CSIHnMCTL2.CSIHnBTST} = 1$ | $\text{CSIHnMCTL2.CSIHnBTST} = 1$ |
| | Slave | Incoming clock from the master | Incoming clock from the master |

14.5.8 Data Length Selection

14.5.8.1 Data Length Between 2 and 16 bits

The length of a data packet is selectable for each chip select signal from 2 to 16 bits using $\text{CSIHnCFGx.CSIHnDLSx}[3:0]$. The examples below show the communication with MSB first ($\text{CSIHnCFGx.CSIHnDIRx} = 0$).

Data length = 16 bits ($\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 0000_{\text{B}}$)

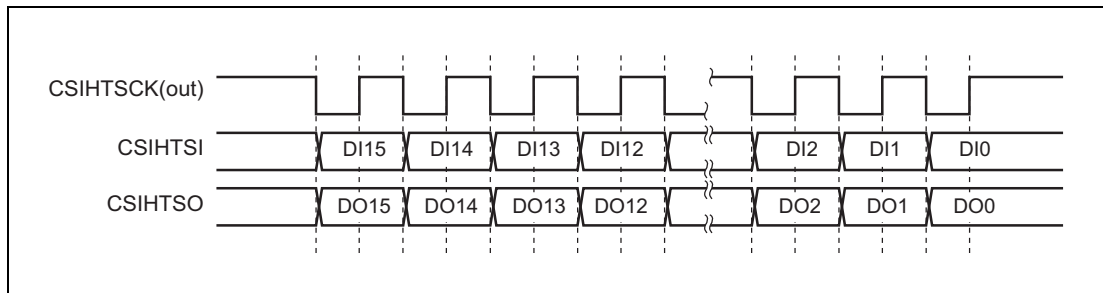


Figure 14.23 16-Bit Data Length, MSB First

Data length = 14 bits ($\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 1110_{\text{B}}$):

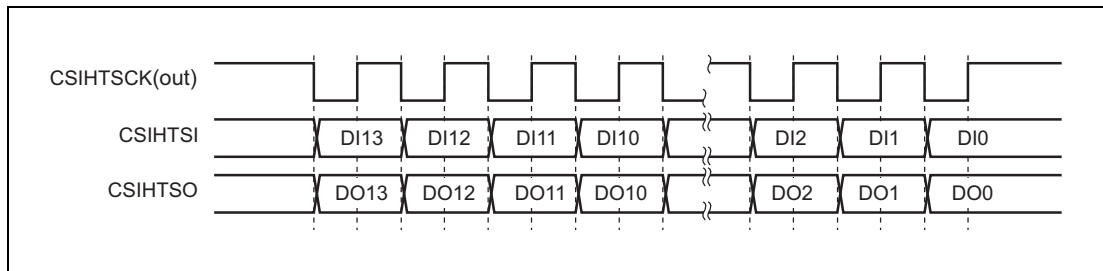


Figure 14.24 14-Bit Data Length, MSB First

14.5.8.2 Data Length Greater than 16 Bits

To transmit/receive data exceeding 16 bits, the extended data length (EDL) feature can be used.

The EDL feature is enabled by setting the CSIHnCTL1.CSIHnEDLE to 1.

The EDL feature works as follows:

- The data must be broken into 16-bit blocks plus remainder. For example, 42-bit data would be broken into two 16-bit blocks plus 10 bits.
- The remainder bit length is set in CSIHnCFGx.CSIHnDLSx[3:0] as “data length.”
- For transmitting the 16-bit blocks, CSIHnTX0W.CSIHnEDL must be set to 1. In this case, the data written to CSIHnTX0W is transmitted as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bit setting.
- The transfer is complete after a block with the specified data length (the remainder of data specified with CSIHnTX0W.CSIHnEDL = 0) has been transmitted.

Example

Example for transmitting 40-bit data (123456789A_H) to CS0:

40 bits are split into two 16-bit blocks and 8 bits.

- Initialize to CSIHnCFG0.CSIHnDLS0[3:0] = 8_H.
- To transmit 123456789A_H with MSB first, write the following sequence to CSIHnTX0W:
 - 20FE 1234_H (CSIHnTX0W.CSIHnEDL = 1)
 - 20FE 5678_H (CSIHnTX0W.CSIHnEDL = 1)
 - 00FE 009A_H (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

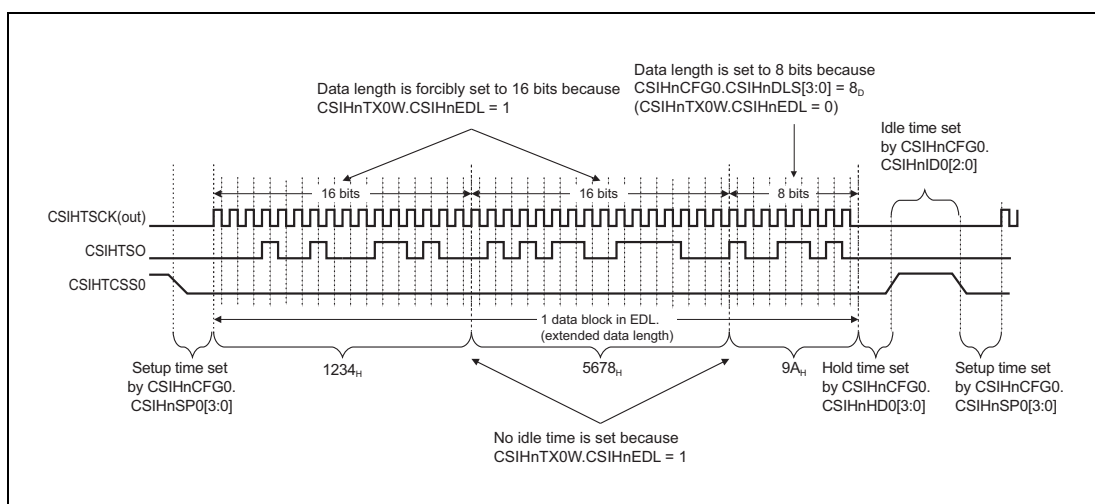


Figure 14.25 EDL Timing Diagram

NOTES

1. Data lengths settings lower than 7 bits are only permitted in combination with EDL mode.
 2. It is not possible to transmit two consecutive data with a data length of less than 7 bits.
 3. If parity is enabled, the parity bit is added after the last bit.
 4. When the extended data length (EDL) feature is used, use the same chip select signal.
 5. If CSIHnTX0W.CSIHnEOJ and CSIHnTX0W.CSIHnEDL are set to 1 at the same time while CSIHnCTL1.CSIHnJE = 1 and CSIHnCTL1.CSIHnEDLE = 1, correct operation is not guaranteed.
 6. Examples for consider the data direction, pay attention to the following example:
 - Data to be transmitted: 123456_H
 - MSB first:
 - Set CSIHnCFGx.CSIHnDIRx = 0
 - Write CSIHnTX0W = 20FE 1234_H (EDL bit = 1)
 - Write CSIHnTX0W = 00FE 0056_H (EDL bit = 0)
 - LSB first:
 - Set CSIHnCFGx.CSIHnDIRx = 1
 - Write CSIHnTX0W = 20FE 3456_H (EDL bit = 1)
 - Write CSIHnTX0W = 00FE 0012_H (EDL bit = 0)
 7. EDL mode cannot be used in receive-only mode of slave mode.
(CSIHnCTL2.CSIHnPRS[2:0] = 111_B, CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1)
-

14.5.9 Serial Data Direction Selection

The serial data direction is selectable for each chip select signal by using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show communication for a data length of 8 bit (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).

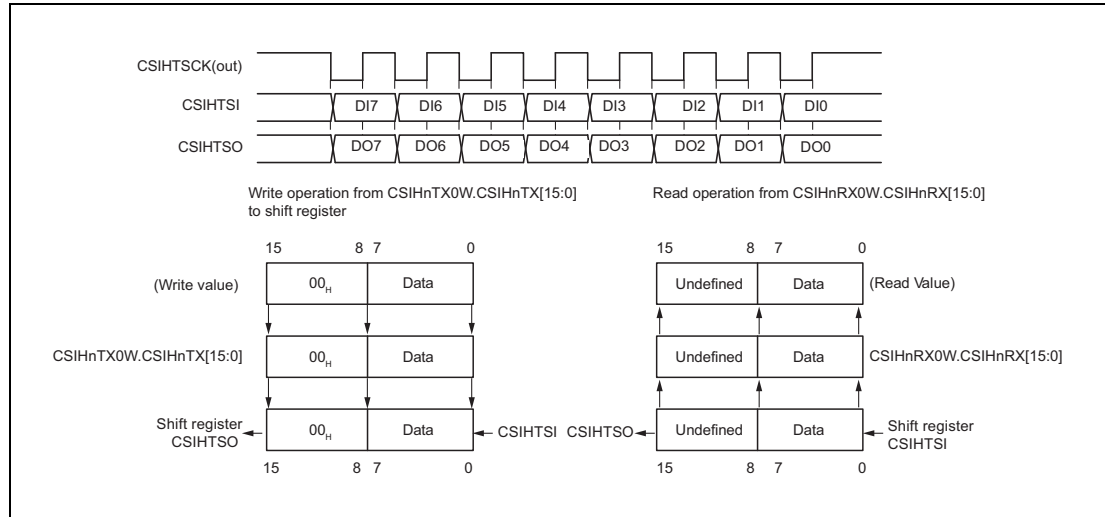


Figure 14.26 Serial Data Direction Select Function - MSB First (CSIHnDIRx = 0)

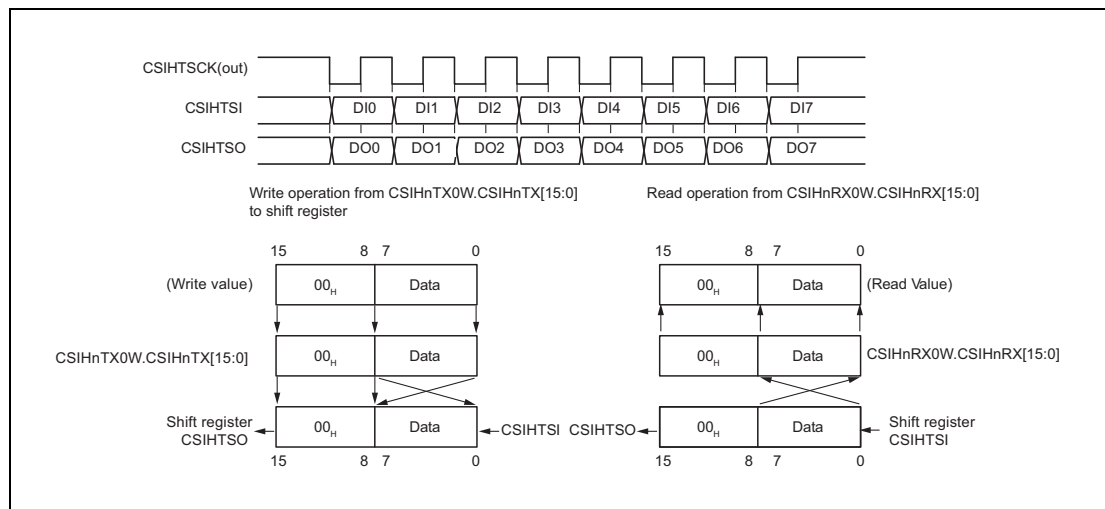


Figure 14.27 Serial Data Direction Select Function - LSB First (CSIHnDIRx = 1)

14.5.10 Slave Select (SS) Function

The SS (slave select) function enables communication between one master and multiple slaves.

In master mode, the master device outputs the slave select signal (CSIHTCSSx) to a slave.

Communication by a device in slave mode is enabled when the slave input select signal ($\overline{\text{CSIHTSSI}}$) is at the low level.

Refer to the **Section 14.5.2, Master/Slave Connections**, for examples of connections using the SS function.

14.5.10.1 Communication Timing Using SS Function

The following figure illustrates the communication signal and timings using the SS function.

In slave mode, the data transfer configuration is determined by the CSIHnCFG0 register.

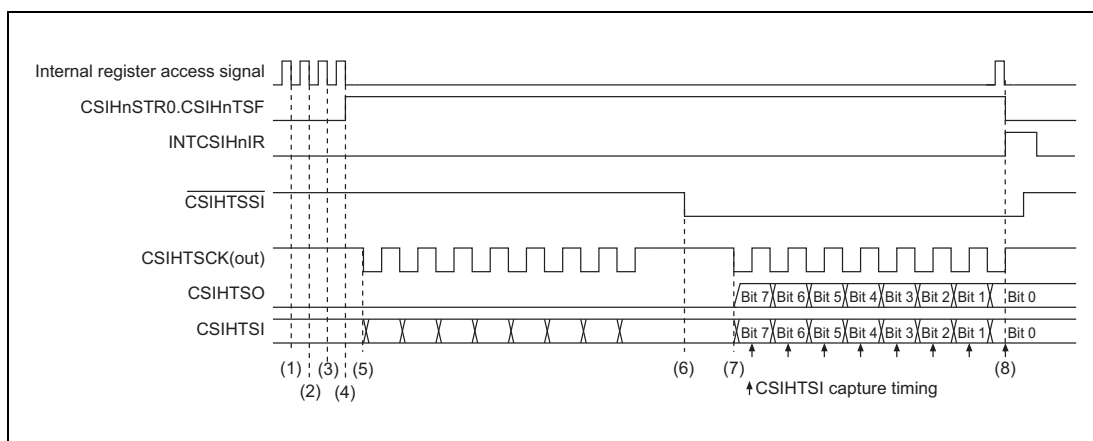


Figure 14.28 Transmission/Reception Timing of Communication Using SS Function

- (1) CSIH enters slave mode by setting CSIHnCTL2.CSIHnPRS[2:0] = 111_B. CSIHnCFG0.CSIHnCKP0 and CSIHnCFG0.CSIHnDAP0 are 0.
- (2) The data length is 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B). The data direction is MSB first (CSIHnCFG0.CSIHnDIR0 = 0).
- (3) Transmit/receive mode is set. (CSIHnCTL0.CSIHnTXE = 1, CSIHnCTL0.CSIHnPWR = 1). Communication start is permitted.
- (4) The transfer status flag CSIHnSTR0.CSIHnTSF is automatically set in direct access mode or FIFO mode when transfer data is written to the CSIHnTX0W or CSIHnTX0H transmission register.
- (5) As long as signal $\overline{\text{CSIHTSSI}}$ is high, transmission/reception is not started, even if an external transmission clock CSIHTSCK is supplied. Input to CSIHTSI is ignored.
- (6) Changing $\overline{\text{CSIHTSSI}}$ to low level indicates that CSIHTSO is enabled and ready for transmission.
- (7) As soon as the external clock signal CSIHTSCK appears, the slave transmits data to CSIHTSO and simultaneously captures data from CSIHTSI.
- (8) Interrupt INTCSIHnIR indicates that the reception is complete. The CSIHnRX0W/H register can be read.

14.5.10.2 CSIHTSSO Operation

| CSIHnPWR | CSIHnTXE | CSIHnRXE | CSIHnSSE | CHISTSSO |
|----------|----------|----------|----------|----------------------------------|
| 0 | — | — | — | H |
| 1 | — | — | 0 | H |
| | 0 | — | 1 | H |
| | 1 | — | 1 | Reversed value of CSIHTSSI level |

The CSIHTSSO pin is a signal to control the I/O function of the chip's SO pin in case of using the SS function.

The CSIHTSSO pin is enabled when the CSIHTSSO pin is "High" (the chip's SO pin is being driven).

The CSIHTSSO pin is disabled when the CSIHTSSO pin is "Low" (the chip's SO pin is not being driven).

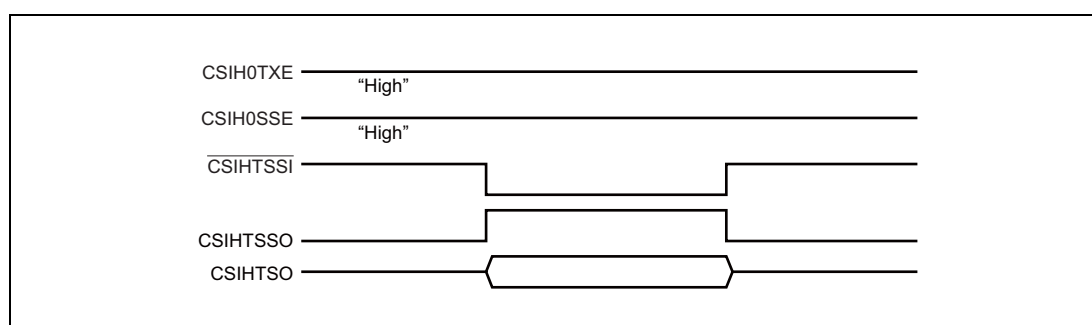


Figure 14.29 Operation of CSIHTSSO

CAUTION

If $\overline{\text{CSIHTSSI}}$ pin is changed during communication ($\text{CSIHnSTR0.CSIHnTSF} = 1$), current communication is not guaranteed.

14.5.11 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIHnCTL1.CSIHnHSE. For handshake, the signals CSIHTRYI and CSIHTRYO are used.

The timing of transitions to the busy state depends on the setting of the data phase selection bit, CSIHnCFGx.CSIHnDAPx.

14.5.11.1 Slave Mode

When the slave becomes busy while CSIHnCTL1.CSIHnHSE = 1, the CSIHTRYO signal outputs low level 0. This state is produced in the following two cases:

1. When the next transmission data is not ready:
When the slave is set to transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1), CSIHTRYO outputs low level (busy state) in the following conditions.

Table 14.45 Memory Mode and Transmission State of Slave

| Memory Mode | Transmission State of Slave |
|---------------------------|--|
| Direct access mode | No next transfer data |
| FIFO mode | No next transfer data (CSIHnSTR0.CSIHnEMF = 1) |
| Dual buffer mode | CSIHnMCTL2.CSIHnBTST is not set to 1 |
| Transmit-only buffer mode | |

The example below describes the waveform in case of an eight-bit data length.

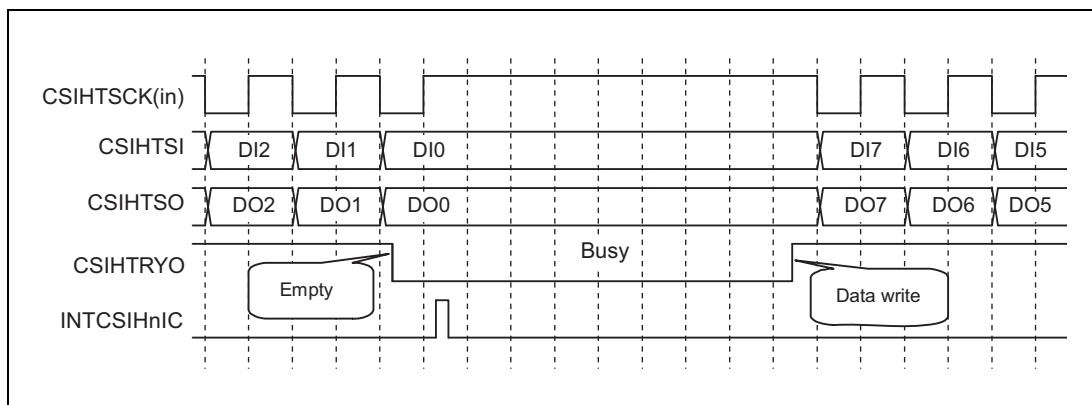


Figure 14.30 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 0)

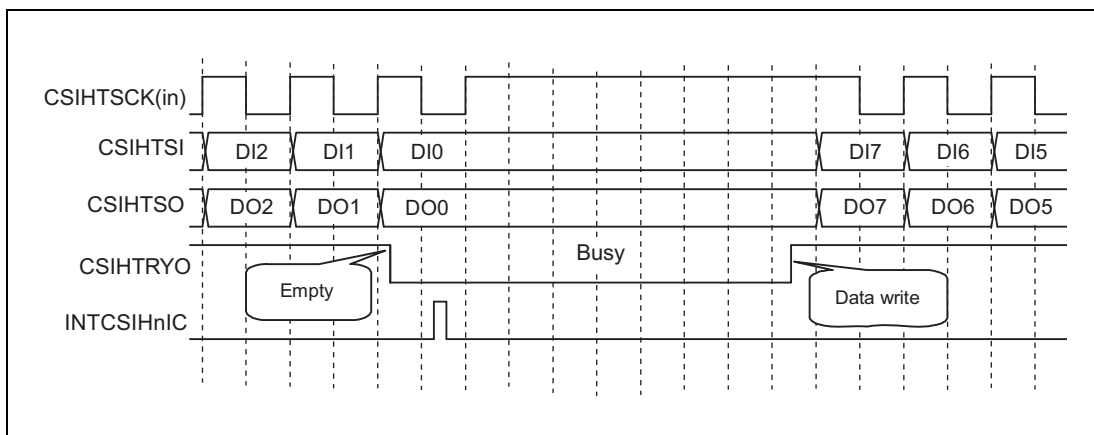


Figure 14.31 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 1)

2. When the receive register is full:

Because the previously received data is still in the CSIHnRX0W or CSIHnRX0H register while the slave is in receive-only or transmit/receive mode (CSIHnCTL0.CSIHnRXE = 1), new data cannot be copied from the shift register to CSIHnRX0W or CSIHnRX0H (CSIHnRX0W or CSIHnRX0H full condition).

When CSIHnCTL0.CSIHnRXE = 1, CSIHTRYO outputs low level (busy state) in the following conditions.

Table 14.46 Memory Mode and Reception State of Slave

| Memory Mode | Reception State of Slave |
|---------------------------|---|
| Direct access mode | CSIHnRX0W or CSIHnRX0H is full |
| FIFO mode | Receive data remaining in the buffer (CSIHnSTR0.CSIHnFLF = 1) |
| Dual buffer mode | No applicable state |
| Transmit-only buffer mode | CSIHnRX0W or CSIHnRX0H is full |

The example below describes the waveform in case of an eight-bit data length.

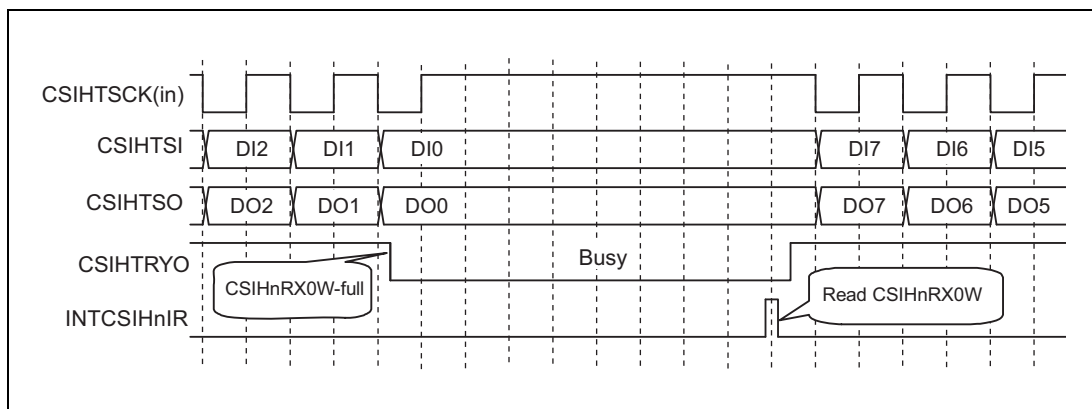


Figure 14.32 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0)

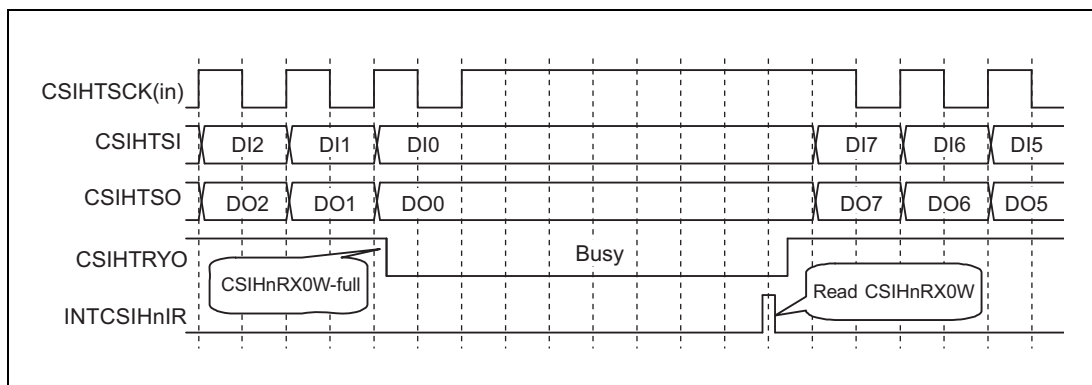


Figure 14.33 Busy Signal from the Slave (CSIHnCFGx.CSIHnDAPx = 1)

14.5.11.2 Master Mode

When the master detects $CSIHTRYI = 0$ while $CSIHnCTL1.CSIHnHSE = 1$, the subsequent transfers are put on hold, the master goes into wait state and suspends the $CSIHTSCK$ clock.

The $CSIHTRYI$ level is checked at each half clock cycle of $CSIHTSCK$.

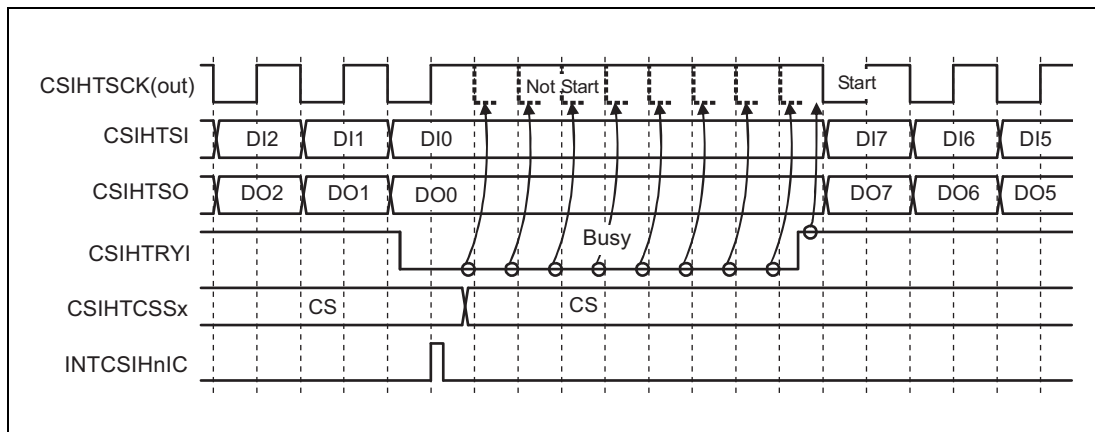


Figure 14.34 Master’s Reaction to CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 0)

$CSIHTRYI$ must be pulled down by the slave before the next transfer starts. When the slave changes the $CSIHTRYI$ signal to low level during data transfer, the serial clock from the master stops at the completion of data transfer.

The master resumes the communication as soon as $CSIHTRYI$ becomes high level (the slave is “ready”).

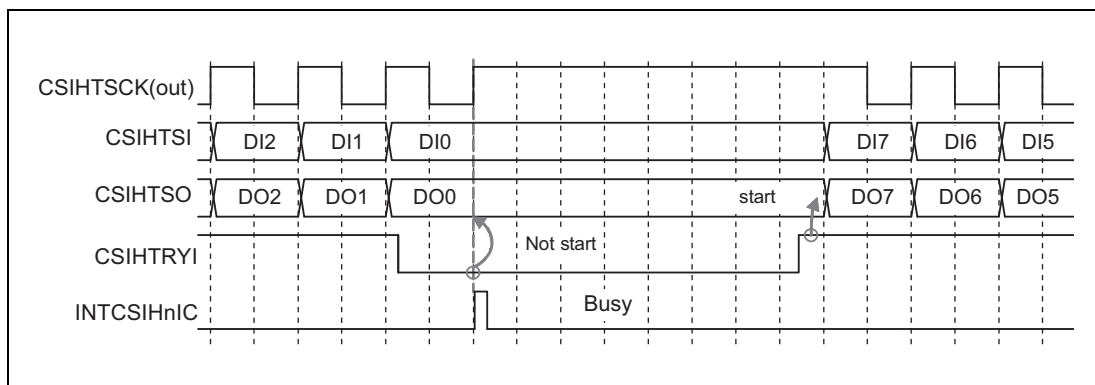


Figure 14.35 Master’s Reaction to CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 1)

CAUTIONS

1. If multiple slaves are connected, the master must detect only the $CSIHTRYI$ signal from the slave it has selected for communication.
2. Even when the $CSIHTRYI$ pin of the master detects the $CSIHTRYO$ signal of the slave during data transfer, communication is not in wait status until data transfer finishes.

14.5.12 Error Detection

CSIH can detect five error types:

- Data consistency check error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Time-out error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency check and time-out errors can be enabled or disabled individually.

If any of these errors are detected, the interrupt request INTCSIHnIRE is generated and the corresponding flags are set.

14.5.12.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically transmitted as output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIHnCTL1.CSIHnDCS bit. When executing a data consistency check, always set CSIHnTXSO to PIPn.PIPn_m = 1. It will not be enabled if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIHnTXSO are read back into a separate shift register.

After completion of the transmission, the transmitted data is compared with the original transmission data.

Mismatch is considered as a data consistency check error and:

- Interrupt INTCSIHnIRE is generated.
- The CSIHnSTR0.CSIHnDCE bit is set.

Additionally, CSIHnRX0W.CSIHnTDCE of data that contains the error is set.

The data consistency check function is illustrated in the following block diagram.

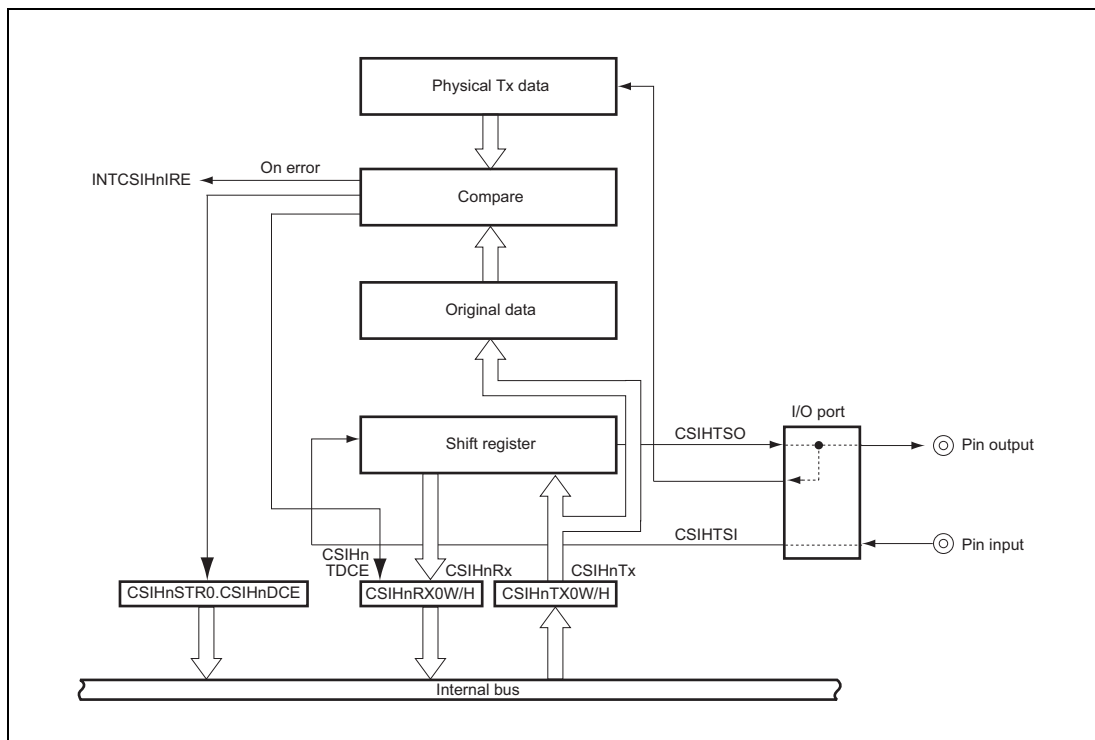


Figure 14.36 Block Diagram of Data Consistency Check Function

14.5.12.2 Parity Check

CSIH can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in $\text{CSIHnCFGx.CSIHnPSx}[1:0]$.

Parity check is enabled if $\text{CSIHnCFGx.CSIHnPSx}[1] = 1$.

The parity bit is checked after a reception is complete. If an parity error occurs, the following happen:

- Interrupt INTCSIHnIRE is generated.
- The CSIHnSTR0.CSIHnPE bit is set.

Additionally, $\text{CSIHnRX0W.CSIHnRPE}$ of data that contains the error is set.

The figure below shows an example.

- Data length is 8 bits.
- The data to be transmitted is 05_{H} and 35_{H} .
- Data direction is LSB first.
- Parity type is odd.

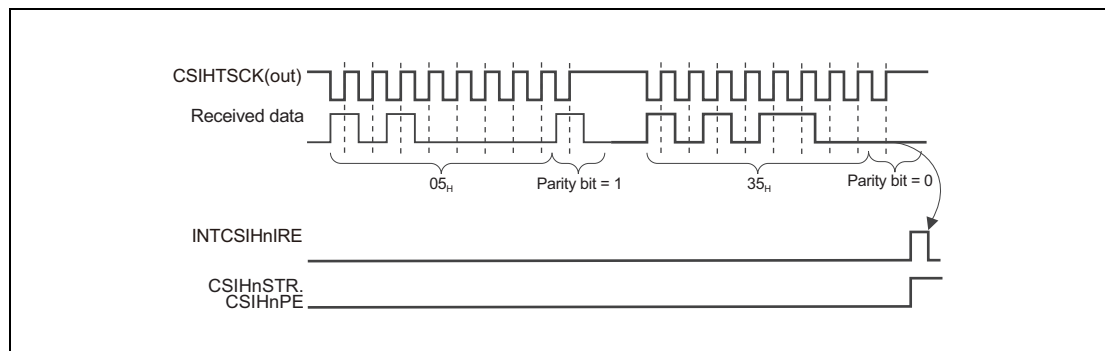


Figure 14.37 Parity Check Example

The parity bit of the first data is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit of the second data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last bit of the data.

14.5.12.3 Time-Out Error

Time-out errors can be checked only in slave FIFO mode of the slave.

This error occurs if neither of the following occurred within a certain period of time:

- Received data in FIFO is read
- FIFO receives data from CSIHTSI

The time-out time is defined in `CSIHnMCTL0.CSIHnTO[4:0]` in units of "8 x transmission clock `CSIHTSCK`". A time-out error occurs when the specified time is exceeded (the time-out time is not detected when `CSIHnMCTL0.CSIHnTO[4:0] = 00000B`).

A dedicated time-out counter is set by the `CSIHnCTL2.CSIHnPRS[2:0]` and `CSIHnBRSy.CSIHnBRS[11:0]` bits.

The `CSIHnBRSy.CSIHnBRS[11:0]` bits is left as set at `000H`, the dedicated time-out counter does not operate.

The dedicated time-out counter measures the time between the last and the next read operation.

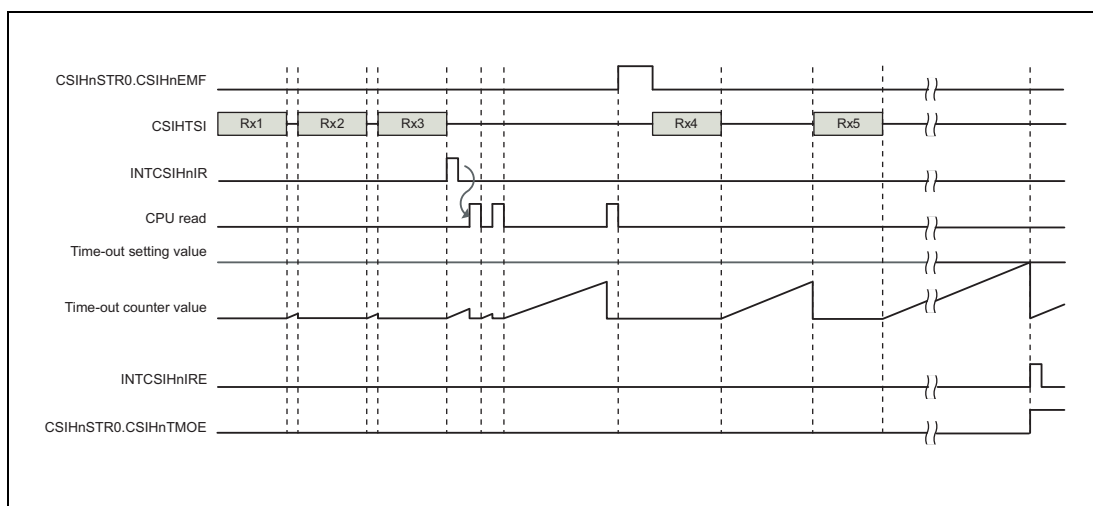


Figure 14.38 Time-Out Check Functional Timing Diagram

The start timing of the time-out counter is as follows:

- When reception is completed
- The CPU reads the received data.
(The counter does not start if the buffer is empty.)
- When a time-out error is detected

After a time-out error is detected, if data is still available in FIFO, the time-out counter restarts.

If the value set by bit `CSIHnMCTL0.CSIHnTO[4:0]` is reached again, the `INTCSIHnIRE` interrupt is output again.

The timeout counter continues to count as long as receive data is not read. To stop the counter, read all received data or set `CSIHnSTCR0.CSIHnPCT` to 1. Note that the pointer is cleared if you set the `CSIHnSTCR0.CSIHnPCT`.

The counter is reset at the following timing:

- Data is read once.

- New data is received.
- A timeout error is detected.
- The CSIHnSTCR0.CSIHnPCT bit is set to 1.

If a timeout error occurs, the following occur:

- Interrupt INTCSIHnIRE is generated.
- The CSIHnSTR0.CSIHnTMOE bit is set.

14.5.12.4 Overflow Error

An overflow error occurs in FIFO mode. It occurs when transmission data is written to the CSIHnTX0W register while the FIFO buffer is filled with received data.

Example

100 data have been transmitted. That is, the FIFO contains 100 received data. The application starts to read the received data.

While the read operation is in progress, the application begins to write another set of 50 transmission data to the FIFO.

However, only 10 received data have been read up to now and 90 data are still in the FIFO.

In this case, only 38 buffers are available for new transmission data. When the CPU tries to write the 39th data, an overflow error occurs.

This is illustrated in the following figure.

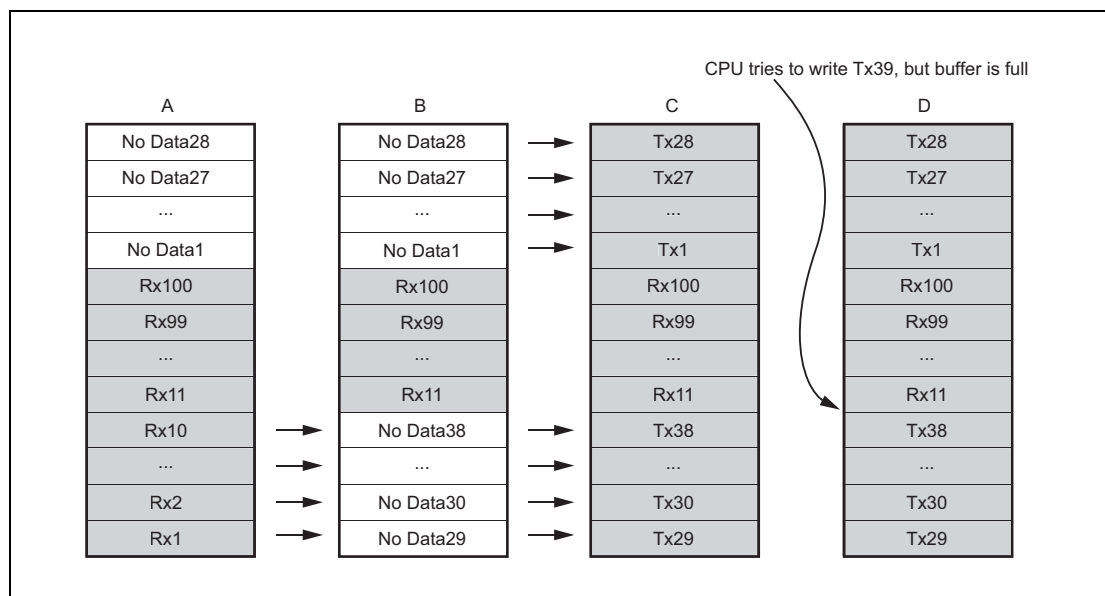


Figure 14.39 FIFO Overview

Data after 39 are discarded. The figure below shows the overflow timing.

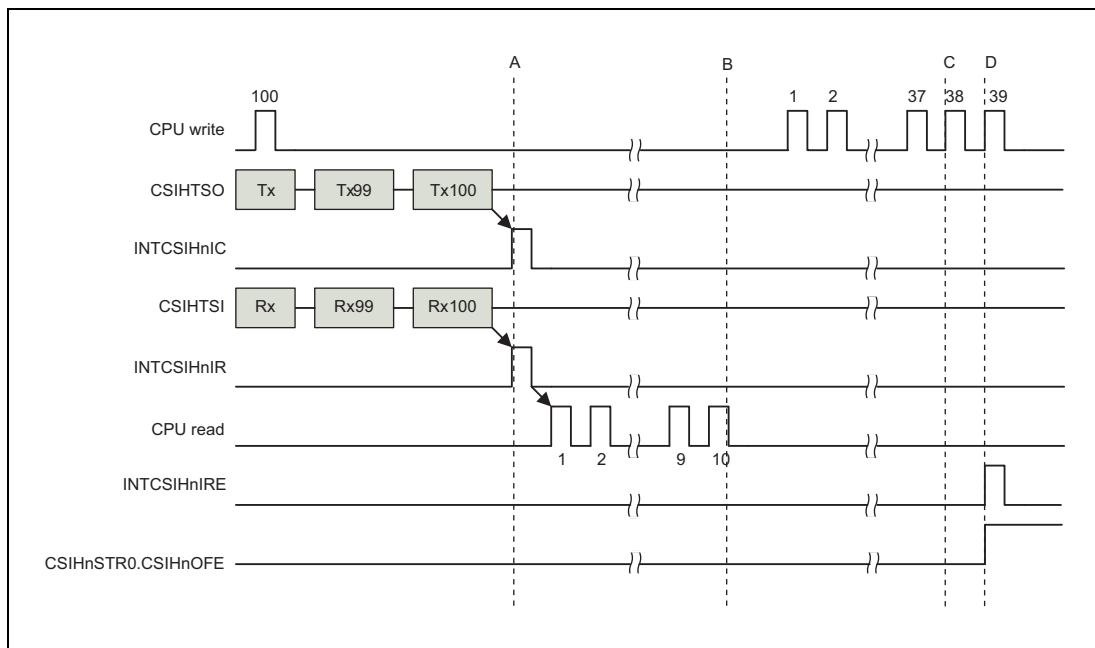


Figure 14.40 FIFO Overflow Timing

In case of overflow error:

- Interrupt INTCSIHnIRE is generated.
- The CSIHnSTR0.CSIHnOFE bit is set.

14.5.12.5 Overrun Error

An overrun error occurs. in direct access, transmit-only buffer, and FIFO modes. It does not occur in dual buffer mode. The overrun error does not occur if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

An overrun error occurs under the two conditions below.

(Error occurrence condition 1)

- When the CPU reads the CSIHnRX0W/H register with no receive data remaining in FIFO mode

(Error occurrence condition 2)

- When CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled) in slave mode
 - When reception is completed with the previous receive data remaining in the CSIHnRX0W/H register in direct access mode or transmit-only buffer mode
 - When reception is completed while the FIFO buffer is full of receive data in FIFO mode

(1) Direct access/transmit-only buffer

In direct access and transmit-only buffer modes, this error occurs when newly received data cannot be transferred from the shift register to the reception register CSIHnRX0W/H without overwriting it. This happens when CSIHnRX0W/H was not read and therefore contains previously received data.

The following figure illustrates the overrun error detection function.

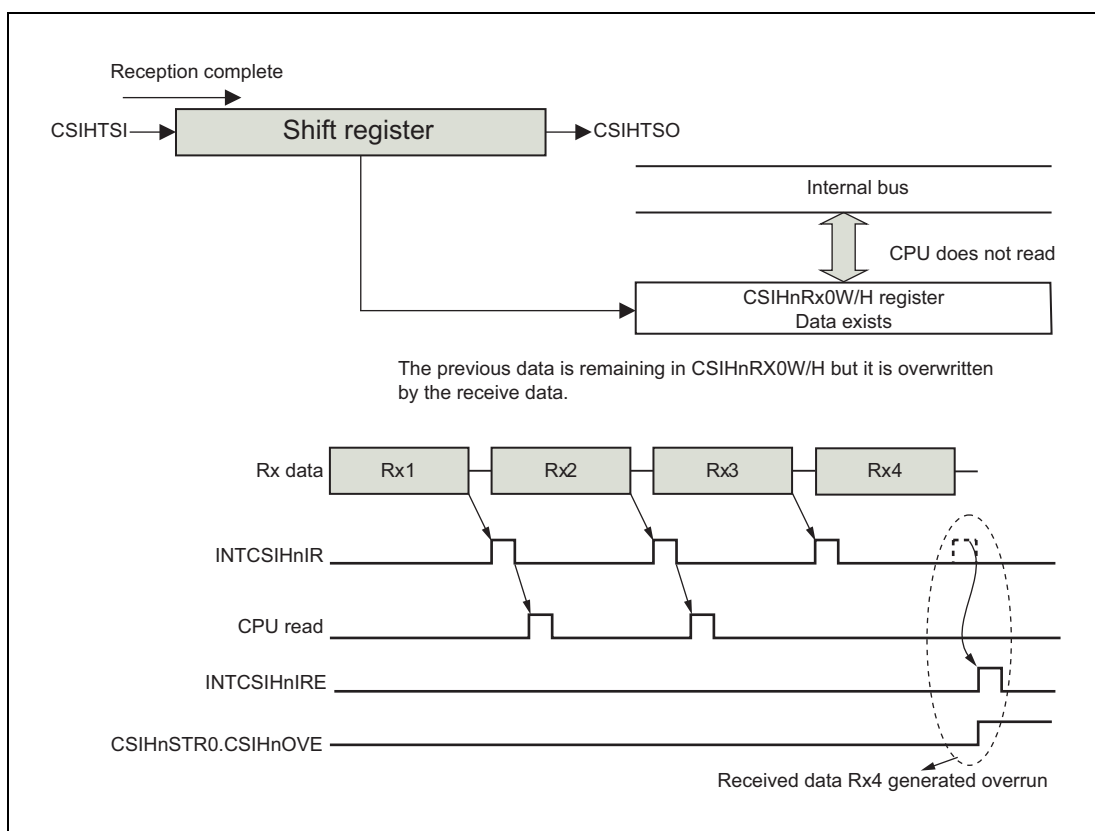


Figure 14.41 Overrun Error Detection in Direct Access and Transmit-Only Buffer Modes

NOTE

An overrun error can be avoided in slave mode by using the handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

(2) FIFO mode

In FIFO mode, this error occurs if:

1. New data is received while the FIFO is full and the old data in the buffer is overwritten.

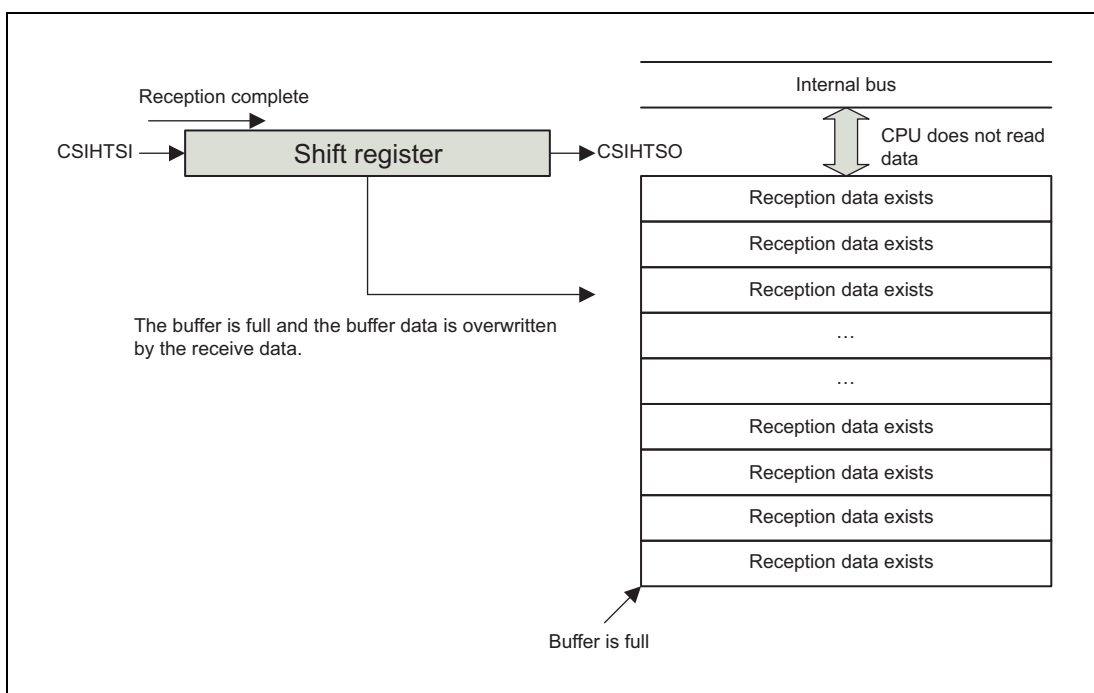


Figure 14.42 Overrun Error Detection in FIFO Mode (FIFO Full)

NOTE

An overrun error can be avoided in slave mode by using the handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

2. The CPU attempts to read non-existent receive data.

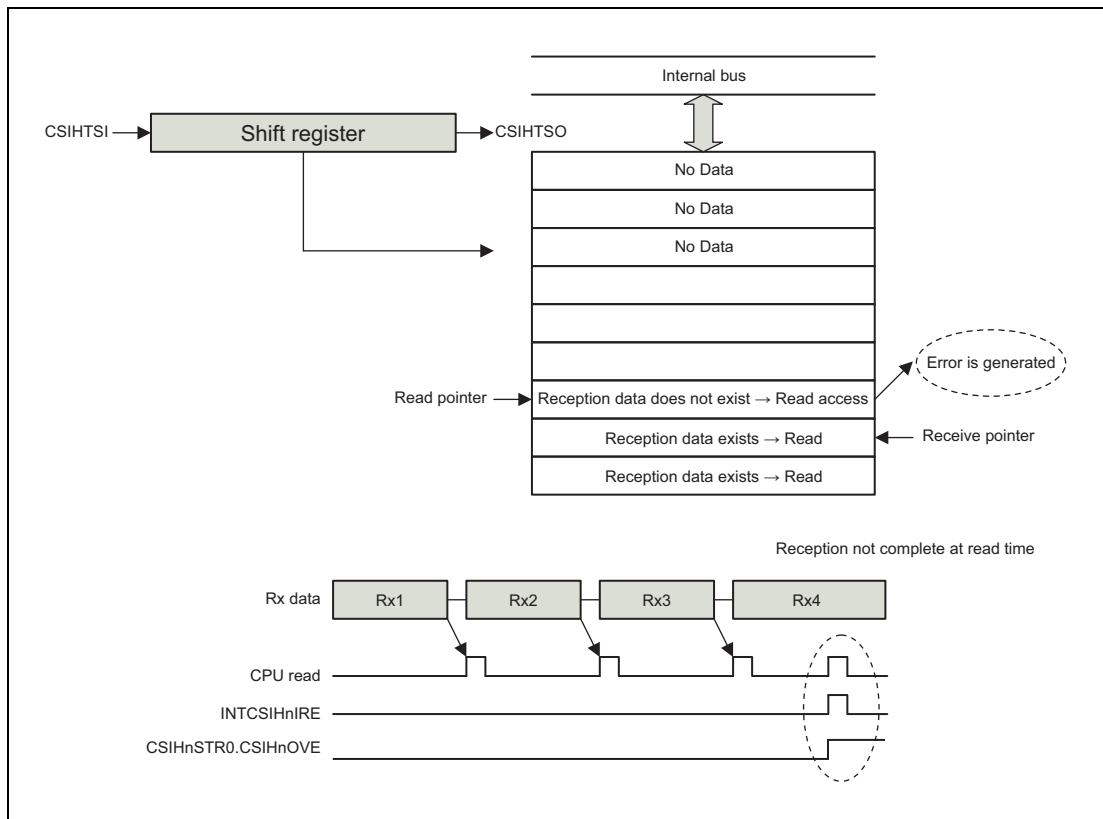


Figure 14.43 Overrun Error Detection in FIFO Mode (No Data)

In case of overrun error:

- Interrupt INTCSIHnIRE is generated.
- The CSIHnSTR0.CSIHnOVE bit is set.
- Receive data is overwritten and communication continues.
(The CPU can read received data after reception is completed.)

For details, see **Section 14.5.11, Handshake Function**.

14.5.13 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIHnCTL1.CSIHnLBM = 1), CSIHTCSSx is fixed to the inactive level (the active level is defined by the CSIHnCTL1.CSIH0CSLx value). The transmit and receive signals are internally connected, as shown in the figures below. The signals CSIHTSCK, CSIHTSO, and CSIHTSI are disconnected from the ports. In addition, the CSIHTSO output level is fixed to low, and CSIHTSCK is set to high level regardless of the CSIHnCFGx.CSIHnCKPx value.

In order to test CSIH, put it in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data. The loopback test does not affect connected devices.

Table 14.47 Pin Output Level when Loopback Mode is Used

| Pin Name | Output Level |
|---------------|---|
| CSIHTSCK(out) | High level |
| CSIHTCSS[7:0] | Inactive level |
| CSIHTSO | Low level (independent of the previous value) |
| Interrupt | Normal function |
| CSIHTRYO | Normal function (low level) |

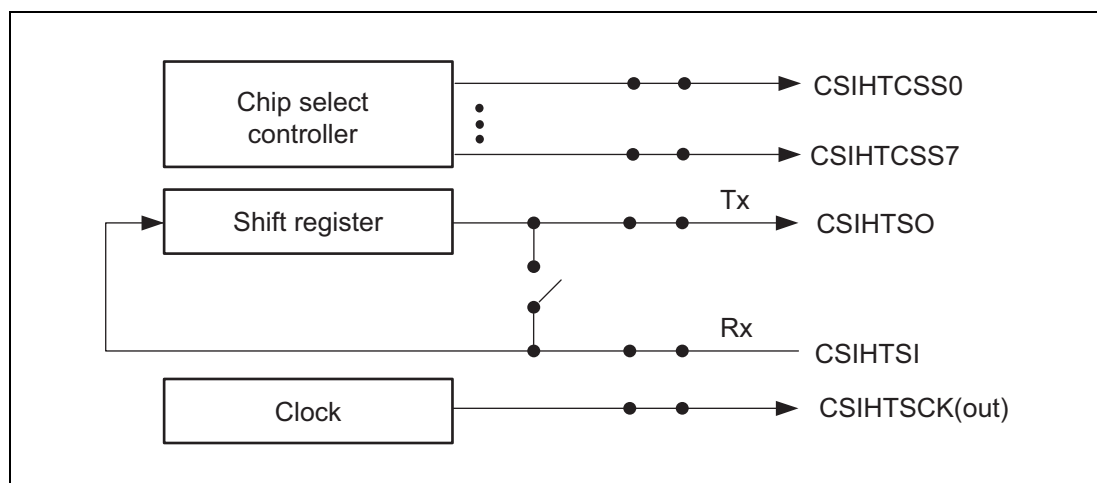


Figure 14.44 Normal Operation

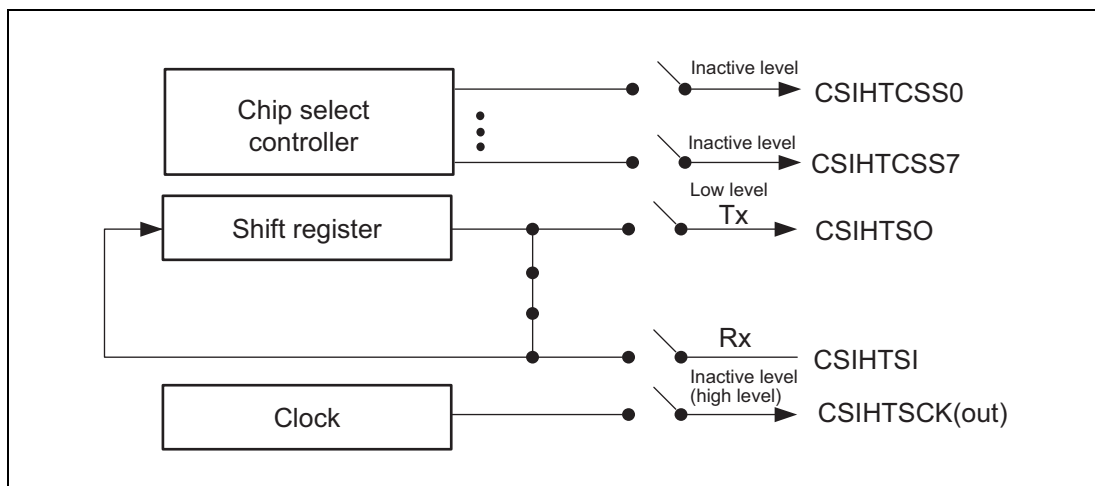


Figure 14.45 Loopback Mode Operation

14.5.14 CPU-Controlled High Priority Communication Function

CSIH has a function to abort low priority communication to perform high priority communication if it receives a high-priority communication request from the CPU while low-priority communication is being used. This function supports transmit-only buffer mode as low priority communication and only direct access mode as high-priority communication.

To enable this function, set $CSIHnCTL1.CSIHnPHE = 1$ and $CSIHnCTL1.CSIHnJE = 1$.

The following figure illustrates CPU-controlled high-priority communication.

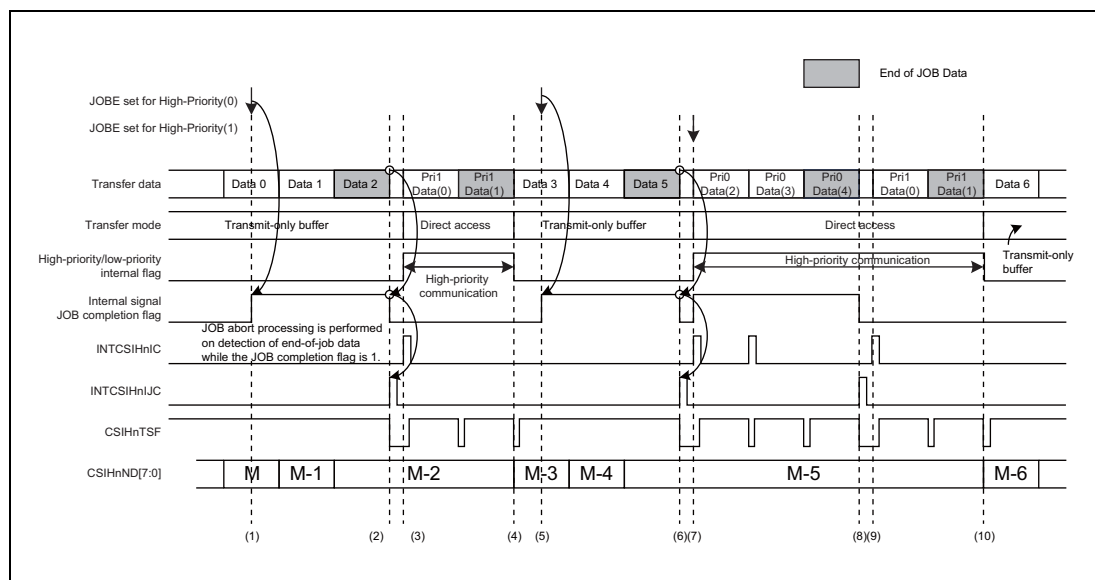


Figure 14.46 Example of CPU-Controlled High-Priority Communication

- (1) By setting $CSIHnCTL0.CSIHnJOBE = 1$ during low-priority communication, start of high-priority communication following end-of-job data is notified, and the internal signal flag is set.
- (2) When end-of-job data is detected, the current low-priority communication is aborted and the $INTCSIHnIJC$ interrupt occurs. An internal signal, the JOB completion flag is cleared due to the communication abort, and memory mode is automatically switched to direct access mode for the subsequent high-priority communication.
- (3) The CPU detects the interrupt and starts communication by writing the first transmission data of high-priority communication to $CSIHnTX0W$ or $CSIHnTX0H$.
- (4) When end-of-job data is detected, communication is aborted. At this time, because the internal signal end-of-job flag is set to 0, the CSIH determines that the next communication is low-priority and switches memory mode to transmit-only buffer mode automatically, and then resumes the aborted low-priority communication.
- (5) Same as (1) above.
- (6) Same as (2) above.
- (7) The CPU detects an interrupt and starts communication by writing the first transmission data of high-priority communication to $CSIHnTX0W$ or $CSIHnTX0H$. The CPU sets $CSIHnCTL0.CSIHnJOBE = 1$ again to notify that the next communication is high-priority.
- (8) When end-of-job data is detected, communication is aborted and the $INTCSIHnIJC$ interrupt is generated. At this time, the CPU determines that the subsequent communication is also high-priority because the internal signal end-of-job flag is 1, and waits for communication to start.
- (9) Same as (3) above.

(10) Same as (4) above.

CAUTION

Memory mode is switched automatically when communication is changed from low priority to high priority (from transmit-only buffer mode to direct access mode) and from high priority to low priority (from direct access mode to transmit-only buffer mode).

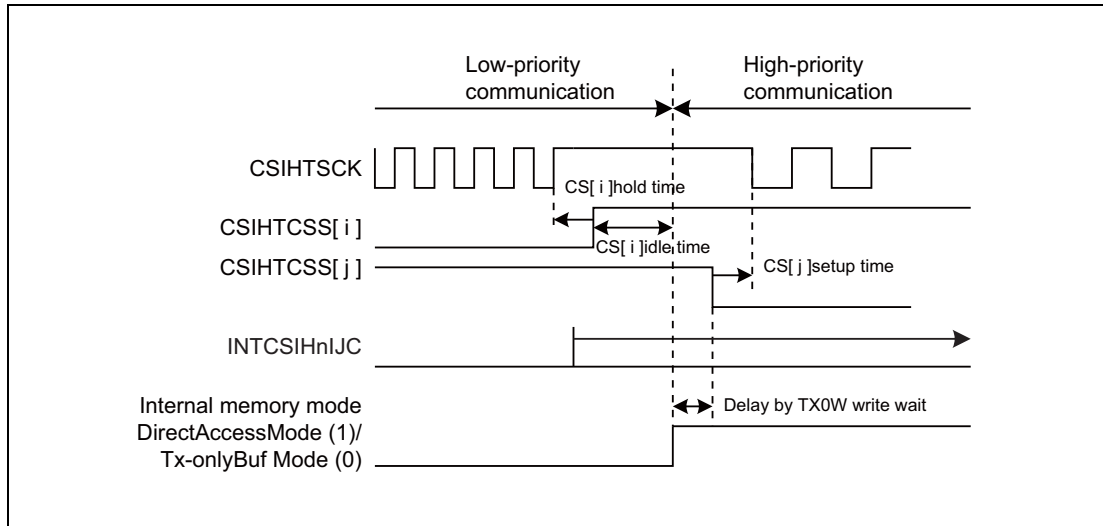


Figure 14.47 Transition from Low-Priority Mode to High-Priority Mode

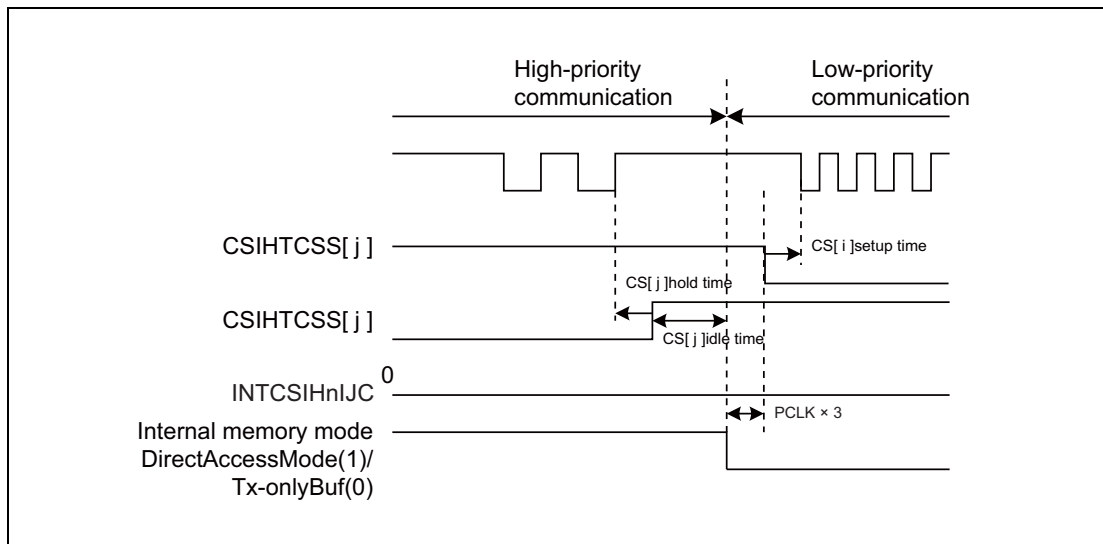


Figure 14.48 Transition from High-Priority Mode to Low-Priority Mode

To correctly switch low-priority communication mode and high-priority communication mode, do not write communication data or manipulate the CSIHnCTL0.CSIHnJOBE bit during the setting prohibited time period.

CSIHnTX0W register write prohibited time period:

- A time period from CSIHnJOBE bit setting for transition to high-priority communication mode until an INTCSIHnJC interrupt is detected

- A time period from the last end-of-job data write of high-priority communication until (CSIHnHPST state = 0) is detected

CSIHnJOBE register write prohibited time period:

- A time period from CSIHnJOBE bit setting for transition to high-priority communication mode until an INTCSIHnIJC interrupt is detected

There is no setting prohibited time period of the CSIHnJOBE bit during the high-priority communication mode. It is also possible to set the CSIHnJOBE bit before writing communication data. For example, when it has been made clear in advance that multiple job data are sent and received in high-priority communication mode, the CSIHnJOBE bit can be set before the first transmission data is written.

CAUTION

If the CSIHnJOBE bit is set immediately before the last high-priority communication ends, operation differs depending on the timing for internally detecting the CSIHnJOBE bit setting.

If the setting of the CSIHnJOBE bit is detected before communication of the last bit is completed, high-priority communication mode continues.

When the CSIHnJOBE bit setting is detected after communication of the last bit has been completed, low-priority communication mode is restored and then end-of-job data of low-priority communication is detected and mode changes again to high-priority communication mode.

14.5.15 Enforced Chip Select Idle Setting

CSIHn units can insert an idle state between the two consecutive transfer data by the setting of CSIHnCFGx.CSIHnIDLx. Detail is as follows.

1. When CSIHnCFGx.CSIHnIDLx = 0
 - If the next CSIHnCSSx is the same as the previous one, an idle state is not inserted and an inter-data time is inserted.
 - If the next CSIHnCSSx is different from the previous one, an idle state is inserted.
2. When CSIHnCFGx.CSIHnIDLx = 1
 - An idle state is always inserted even if a next CSIHnCSSx is not different from the previous one.

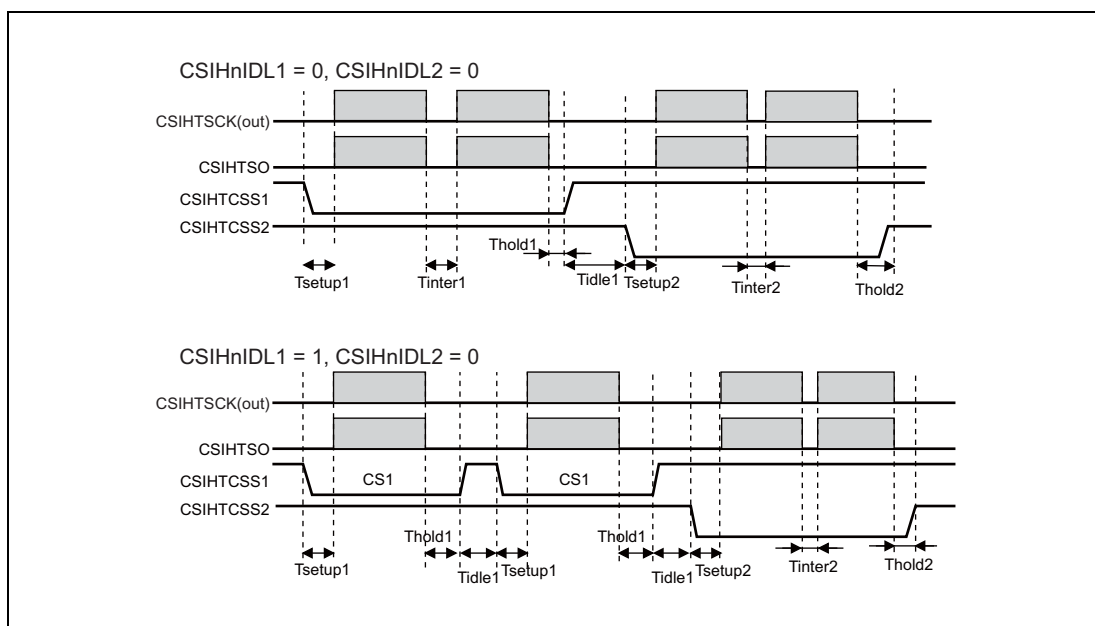


Figure 14.49 Enforced Chip Select Idle Setting Example

CAUTION

If the CPU-controlled high priority communication function is enabled (CSIHnCTL1.CSIHnPHE = 1), when the mode is switched from low priority communication to high priority communication or from high priority communication to low priority communication, an IDLE state is inserted regardless of the setting of CSIHnCFGx.CSIHnIDLx bit.

14.6 Operating Procedures

The examples and procedures below are described according to the memory mode in the following order:

- Direct access mode
- Transmit-only buffer mode
- Dual buffer mode
- FIFO mode

14.6.1 Procedures in Direct Access Mode

Two examples for a master are provided, one with job mode disabled, and the other with job mode enabled.

14.6.1.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedures below is based on the assumption that:

- The transmission data length is 8-bits ($\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 1000_{\text{B}}$).
- Transmission direction is MSB first ($\text{CSIHnCFGx.CSIHnDIRx} = 0$).
- Normal clock and data phase ($\text{CSIHnCFGx.CSIHnCKPx} = 0$, $\text{CSIHnCFGx.CSIHnDAPx} = 0$).
- No interrupt delay ($\text{CSIHnCTL1.CSIHnSIT} = 0$).
- Job mode is disabled ($\text{CSIHnCTL1.CSIHnJE} = 0$).
- Normal INTCSIHnIC interrupt timing ($\text{CSIHnCTL1.CSIHnSLIT} = 0$).
- Direct access mode ($\text{CSIHnCTL0.CSIHnMBS} = 1$).

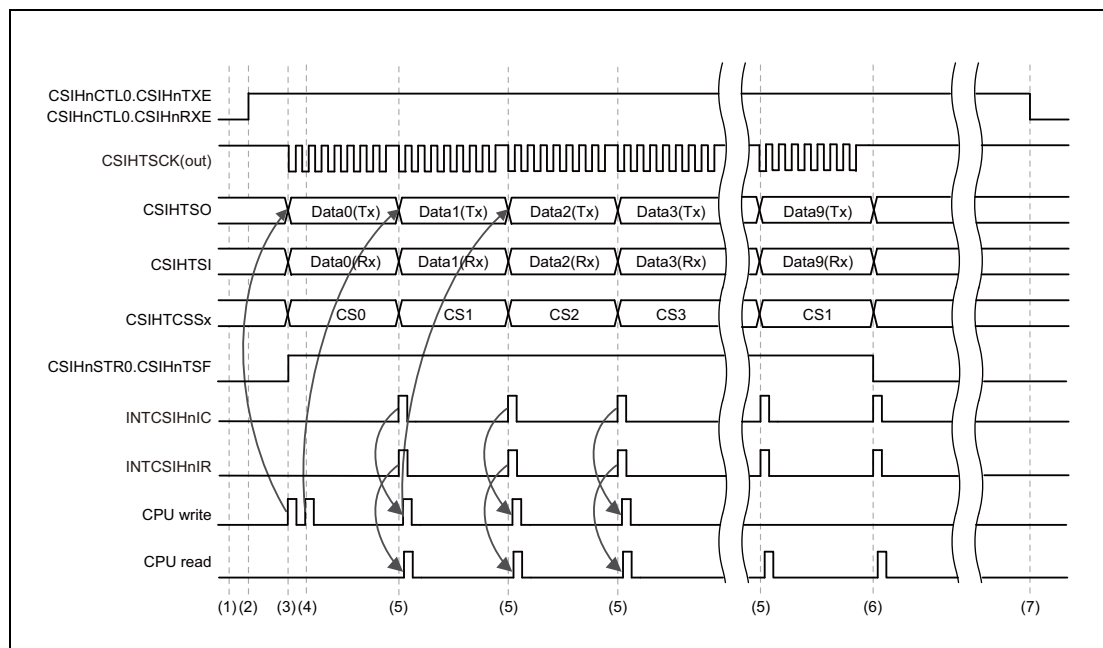


Figure 14.50 Master in Direct Access Mode, $\text{CSIHnCTL1.CSIHnJE} = 0$

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS0 to CS3.
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first transmission data to the transmission register CSIHnTX0W. Within the same write operation, activate CS0. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIHnTX0W. If required, you can change the CS to address a different device. Writing the second data immediately after the first one avoids unnecessary delays between data.
5. Each time data is transmitted and received, INTCSIHnIC and INTCSIHnIR interrupts are generated.
 - INTCSIHnIC indicates that the next data can be written to CSIHnTX0W.
 - INTCSIHnIR indicates that the reception register, CSIHnRX0W must be read.
6. No more write action is required after completion of transmission of data 8. Data 9 (the last data) has been written when a successful transmission interrupt of data 7 was generated. However, reception register CSIHnRX0W must be read after completion of writing data 8 and 9.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

14.6.1.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs, each of them transmits three data.

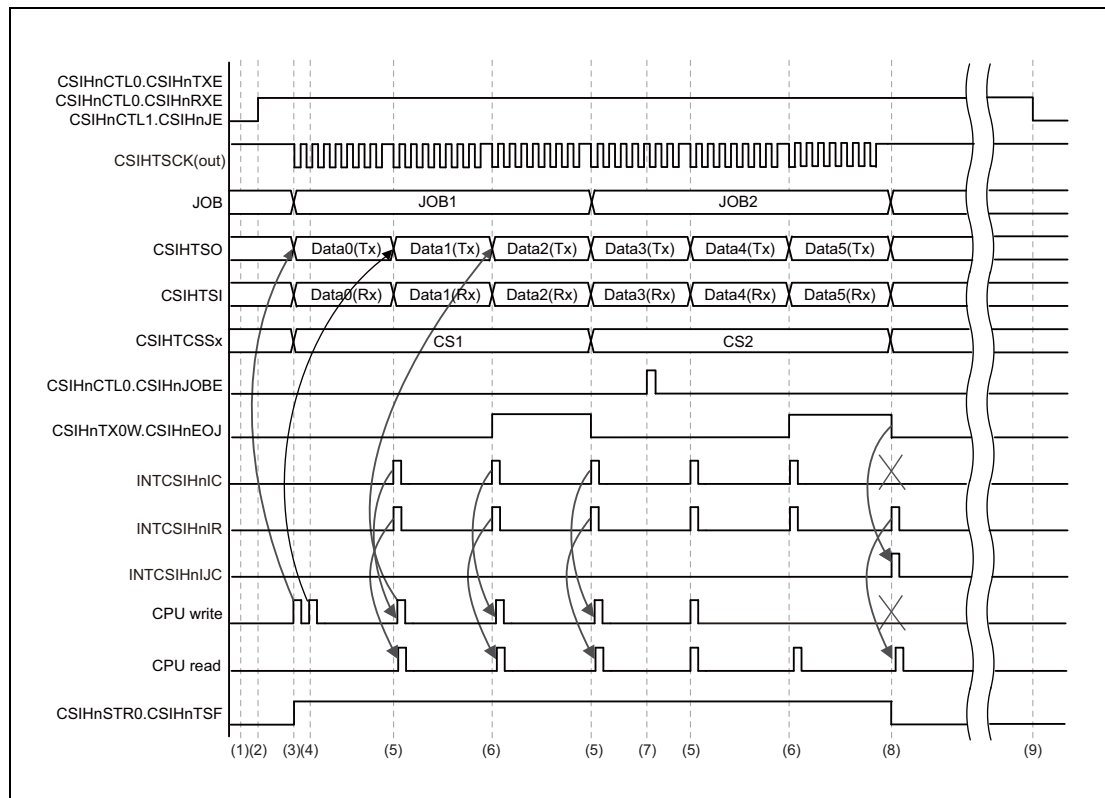


Figure 14.51 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1 and CS2.
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits the reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first transmission data to the transmission register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
The CSIHnSTR0.CSIHnTSF flag indicates that communication is in progress.
4. Write the second data to CSIHnTX0W. Writing the second data immediately after the first one avoids unnecessary delays between data.
5. Each time data is transmitted and received, INTCSIHnIC and INTCSIHnIR interrupts are generated.
 - INTCSIHnIC indicates that the next data can be written to CSIHnTX0W.
 - INTCSIHnIR indicates that the reception register CSIHnRX0W must be read.
6. Setting CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is transmitted.
7. By setting CSIHnCTL0.CSIHnJOB2 = 1, communication is forced to stop at the end of the current job (JOB2).
8. After the forced stop of communication, the interrupt request INTCSIHnIC is replaced by INTCSIHnIJC. INTCSIHnIR is generated as usual.
The interrupt request INTCSIHnIJC indicates that the communication was forcibly stopped at the end of the current job.
The interrupt request INTCSIHnIC is not generated. Additionally, the transmission data available in the CSIHnTX0W register is not transmitted.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

To start another transmission without stopping communication, perform steps 3 and later.

14.6.2 Procedures in Transmit-Only Buffer Mode

Two examples for a master is provided, one with job mode disabled, and the other one with job mode enabled.

14.6.2.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Number of data: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

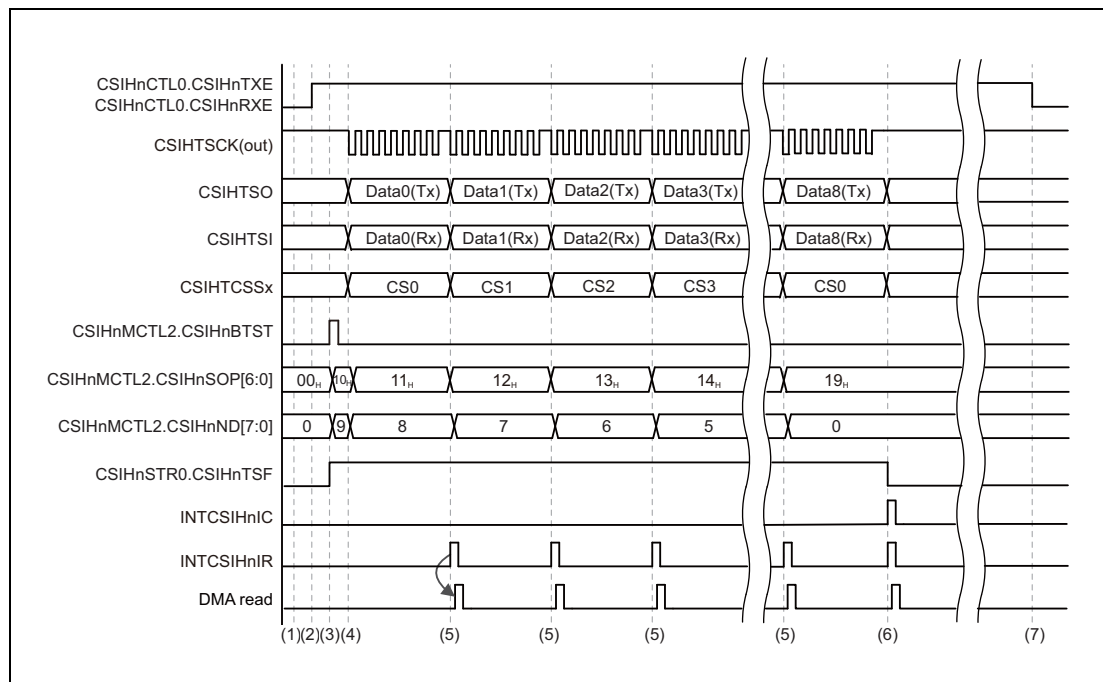


Figure 14.52 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The procedure of writing the data into the buffer is not described here.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHnCSS0 to CSIHnCSS3.
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
Set the memory mode by CSIHnMCTL0.CSIHnMMS[1:0].
Set CSIHnMCTL0.CSIHnMMS[1:0] to 10_B (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission) and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the transmission pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. An interrupt request INTCSIHnIR is generated each time a piece of data is received. INTCSIHnIR indicates that the reception register, CSIHnRX0W must be read.
6. When all transmissions are completed, the interrupt request, INTCSIHnIC is generated.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

14.6.2.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- Number of data: 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

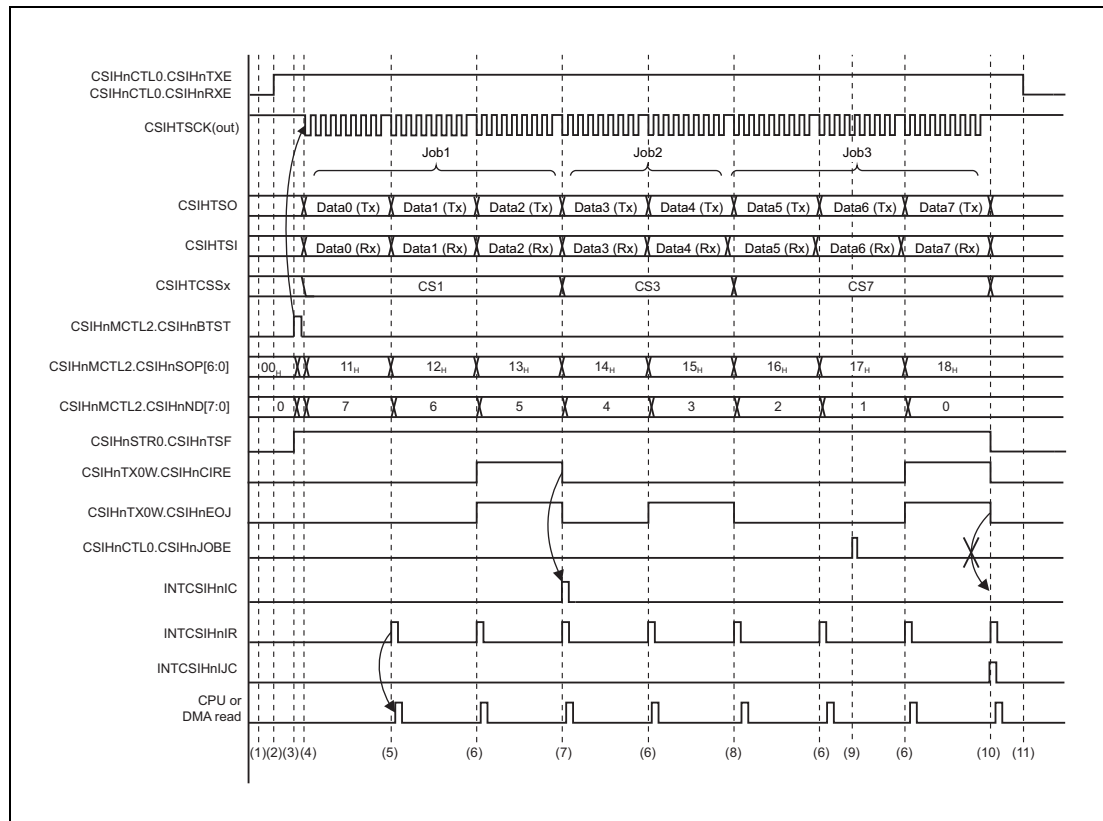


Figure 14.53 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The procedure of writing the data into the buffer is not described here.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1, CS3, and CS7.
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
Set the memory mode by CSIHnMCTL0.CSIHnMMS[1:0].
Set CSIHnMCTL0.CSIHnMMS[1:0] to 10_B (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the transmission pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. An interrupt request INTCSIHnIR is generated each time a piece of data is received. INTCSIHnIR indicates that the reception register CSIHnRX0W must be read.
6. The CSIHnTX0W.CSIHnEOJ = 1 setting indicates that the last data of the current job is transmitted.
7. The interrupt request INTCSIHnIC is generated. INTCSIHnIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was transmitted with CSIHnTX0W.CSIHnCIRE = 1.
8. The INTCSIHnIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CHABnEOJ = 1) was transmitted with CSIHnTX0W.CSIHnCIRE = 0.
9. By setting CSIHnCTL0.CSIHnJOB3 = 1, communication is forced to stop at the end of JOB3.
10. After the forced stop of communication, interrupt requests INTCSIHnIJC and INTCSIHnIR are generated at the end of job3.
The INTCSIHnIJC interrupt request indicates a forced stop of communication at the end of the current job.
The INTCSIHnIC interrupt request is not generated because the INTCSIHnIJC interrupt request is generated instead of the INTCSIHnIC interrupt request. Additionally, the transmission data available in the CSIHnTX0W register is not transmitted.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

14.6.3 Procedures in Dual Buffer Mode

The following examples are provided: those with job mode enabled and disabled during master mode, and one with job mode disabled during slave mode.

14.6.3.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CHIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Number of data: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

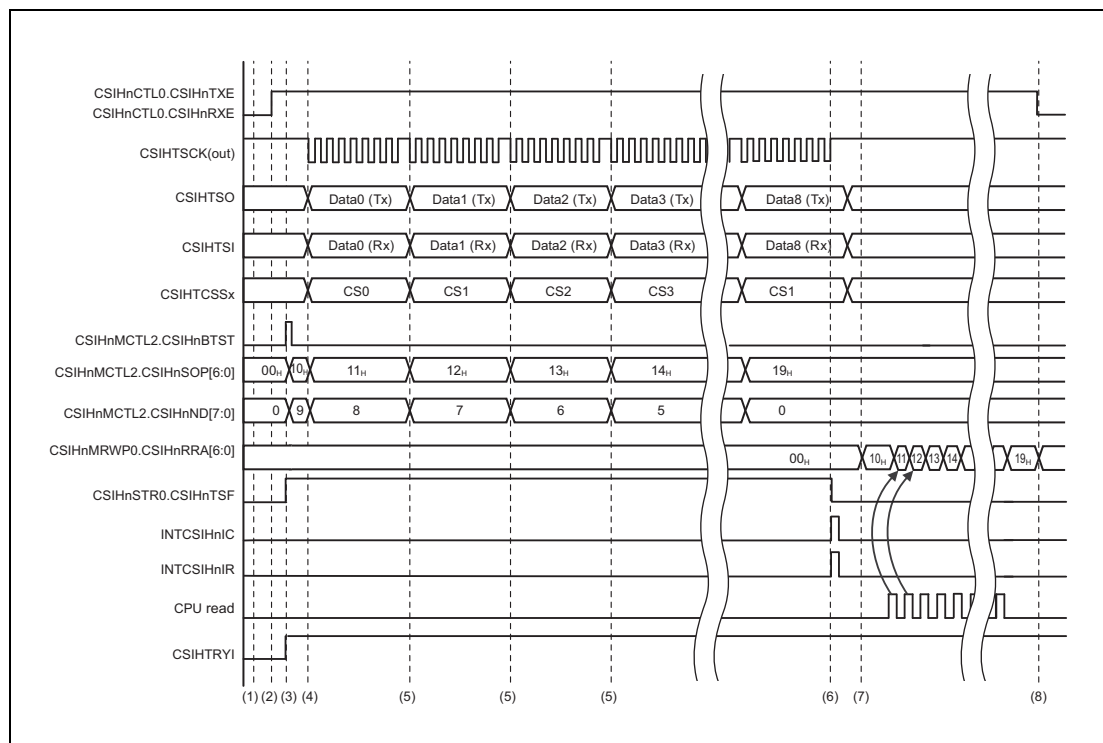


Figure 14.54 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The procedure of writing the data into the buffer is not described here.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. The example uses chip select signals CSIHnCSS0 to CSIHnCSS3.
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
Set the memory mode by CSIHnMCTL0.CSIHnMMS[1:0].
Set CSIHnMCTL0.CSIHnMMS[1:0] to 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits the reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received. The interrupt requests INTCSIHnIC and INTCSIHnIR are not generated.
6. When the last data is transmitted and received, interrupt requests INTCSIHnIC and INTCSIHnIR are generated.
The CPU starts to read the received data from the receive buffer.
7. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0] (CSIHnMRWP0.CSIHnRRA[6:0] is set to 10_H by the software in this figure). These bits are incremented each time a piece of data is read.
8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

14.6.3.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- Number of data: 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 00_H (CSIHnMCTL2.CSIHnSOP[6:0] = 00_H).
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

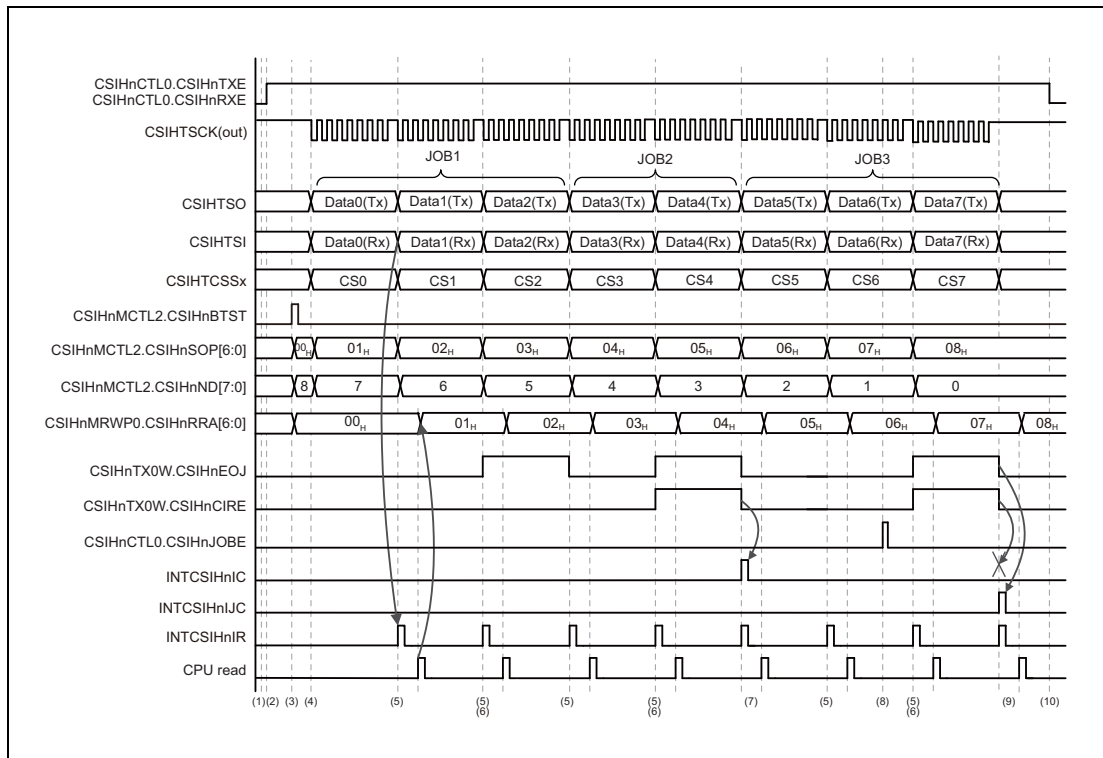


Figure 14.55 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The procedure of writing the data into the buffer is not described here.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. The example uses chip select signals CS0 to CS7.
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
Set the memory mode by CSIHnMCTL0.CSIHnMMS[1:0].
Set CSIHnMCTL0.CSIHnMMS[1:0] to 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission. This is repeated until the last data is transmitted/received.
5. An interrupt request INTCSIHnIR is generated each time a piece of data is received.
The INTCSIHnIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was transmitted with CSIHnTX0W.CSIHnCIRE = 0.
6. CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is transmitted.
7. The INTCSIHnIC interrupt request is generated. INTCSIHnIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was transmitted with CSIHnTX0W.CSIHnCIRE = 1.
8. By setting CSIHnCTL0.CSIHnJOB3 = 1, communication is forced to stop at the end of JOB3.
9. After the forced stop of communication, interrupt requests INTCSIHnIJC and INTCSIHnIR are generated at the end of JOB3.
The INTCSIHnIJC interrupt request indicates a forced stop of communication at the end of the current job.
The INTCSIHnIC interrupt request is not generated because the INTCSIHnIJC interrupt request is generated instead. Additionally, the transmission data available in register CSIHnTX0W is not transmitted.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

14.6.3.3 Transmit/Receive in Slave Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCTL1.CSIHnCKR = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Number of data: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- The handshake function is enabled. (CSIHnCTL1.CSIHnHSE = 1)

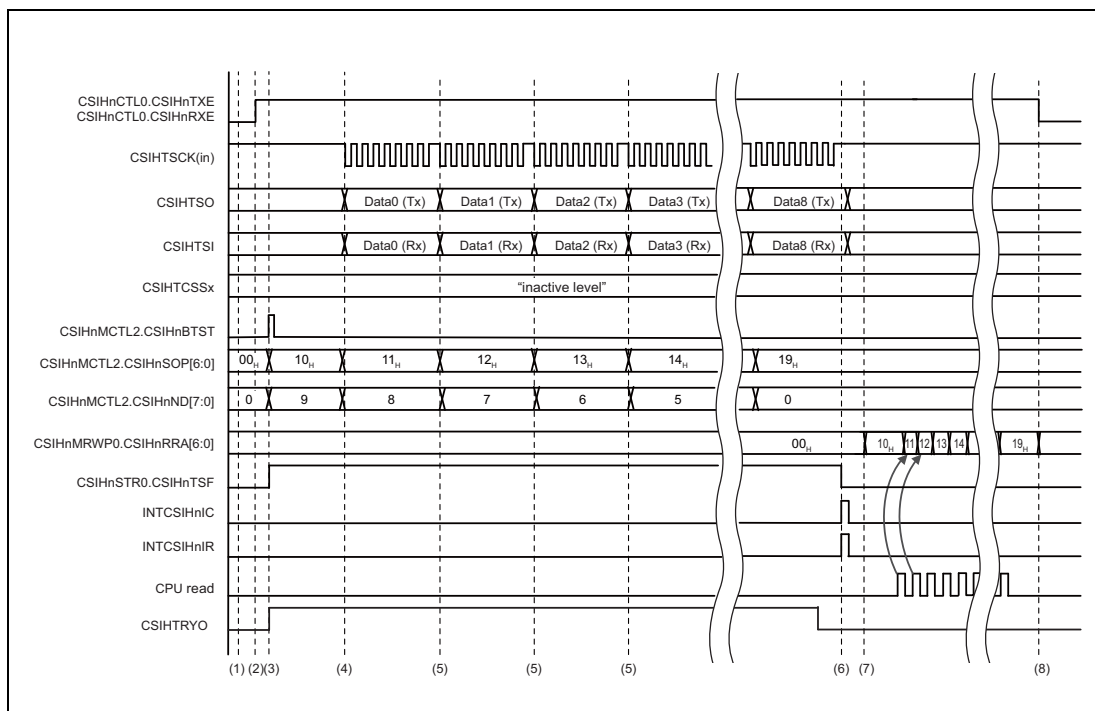


Figure 14.56 Slave in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The procedure of writing the data into the buffer is not described here.

Procedure:

1. Configure the communication protocol in register CSIHnCFG0.
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
Set the memory mode by CSIHnMCTL0.CSIHnMMS[1:0]
Set CSIHnMCTL0.CSIHnMMS[1:0] to 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Specify the transfer start address by setting the CSIHnMCTL2.CSIHnSOP[6:0] bits and the number of data by setting the CSIHnMCTL2.CSIHnND[7:0] bits.
Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started when the input clock is received from the master. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.
The interrupt requests INTCSIHnIC and INTCSIHnIR are not generated because transmission data is sent from the buffer, and received data is stored in the buffer.
6. When the last data is transmitted and received, interrupt requests INTCSIHnIC and INTCSIHnIR are generated.
The CPU starts to read the received data that is stored in the receive buffer.
7. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0] (CSIHnMRWP0.CSIHnRRA[6:0] is set to 10_H by the software in this figure). These bits are incremented each time a piece of data is read.
8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

14.6.4 Procedures in FIFO Mode

Two examples for a master is provided, one with job mode disabled, the other one with job mode enabled.

14.6.4.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

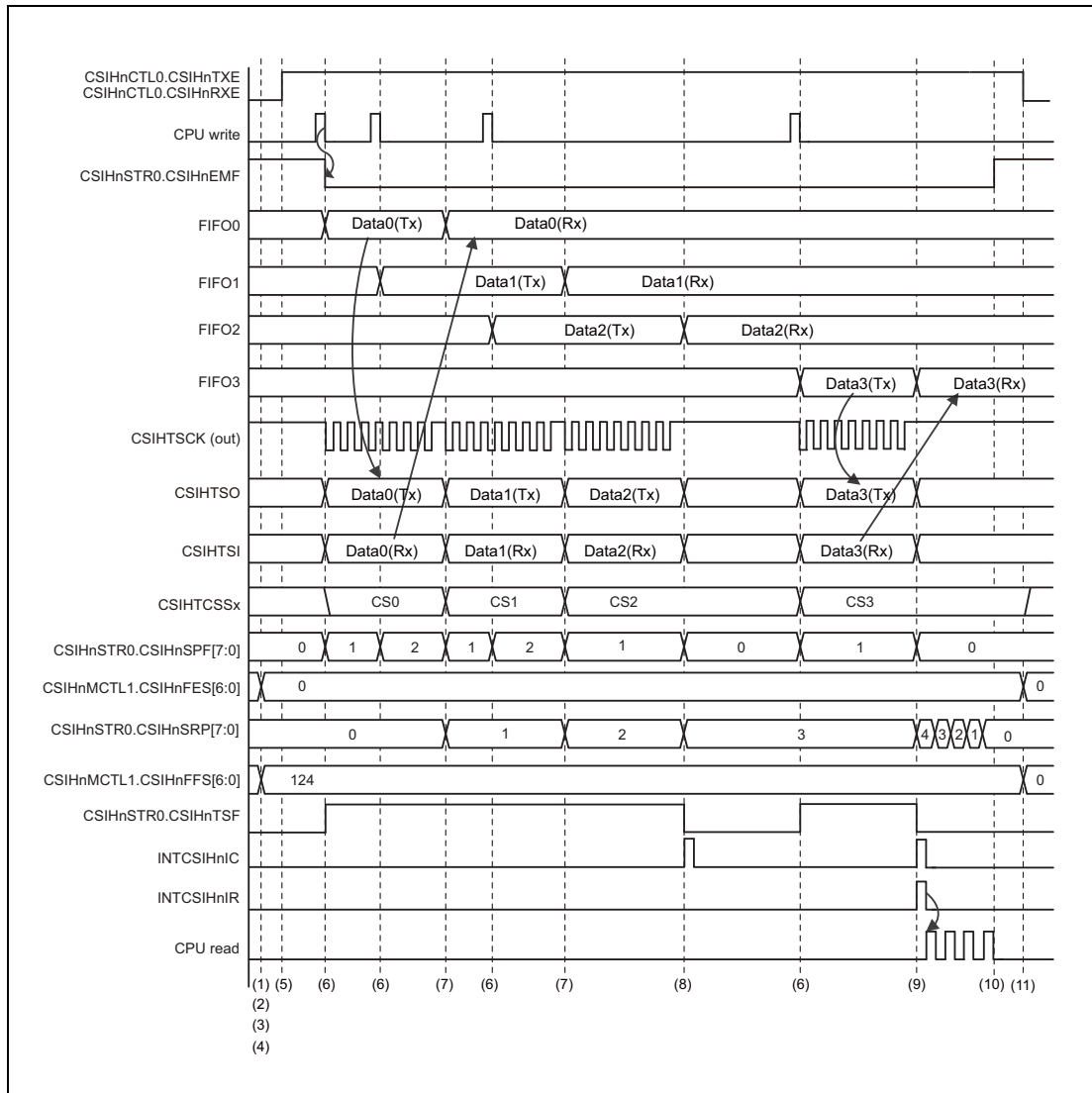


Figure 14.57 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communications protocol in the CSIHnCFGx register, specify the job mode and master mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2, and select FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] to 00_B. These examples use chip select signals CSIHnCSS0 to CSIHnCSS3.
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF is set to 0, CSIHnSTR0.CSIHnEMF is set to 1, and CSIHnSTR0.CSIHnSPF[7:0] is set to 00_H respectively.
4. With CSIHnMCTL1.CSIHnFES[6:0], specify the conditions for the INTCSIHnIC interrupt output. With CSIHnFFS[6:0] in the same register, specify the conditions for the INTCSIHnIR interrupt output.
5. Set CSIHnCTL0.CSIHnPWR is set to 1 (enables the clock), CSIHnTXE is set to 1 (permits transmission), and CSIHnRXE is set to 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
6. Write the transmission data to the transmission register CSIHnTX0W. When the transmission data becomes available, transmission starts automatically.
Make sure that CSIHnSTR0.CSIHnEMF is set to 0.
7. The current transmission is completed. The interrupt request, INTCSIHnIC is not generated because CSIHnFES[6:0] is not CSIHnSPF[7:0].
8. The interrupt request, INTCSIHnIC is generated because CSIHnFES[6:0] is CSIHnSPF[7:0].
9. When CSIHnFFS[6:0] becomes 128-CSIHnSRP[7:0], the interrupt request, INTCSIHnIR is generated. Since CSIHnFES[6:0] = CSIHnSPF[7:0], an interrupt request INTCSIHnIC is generated. After the interrupt is generated, the CPU starts reading the received data stored in the receive buffer.
10. When the CPU has read the received data stored in the receive buffer, CSIHnSTR0.CSIHnEMF is set to 1 and the FIFO buffer becomes empty.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize the power consumption of the CSIHn.

14.6.4.2 Transmit/Receive in Mater Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8-bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- JOB1 consists of four data, JOB2 consists of three data, and JOB3 consists of five data.
- Normal INTCSIHnIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

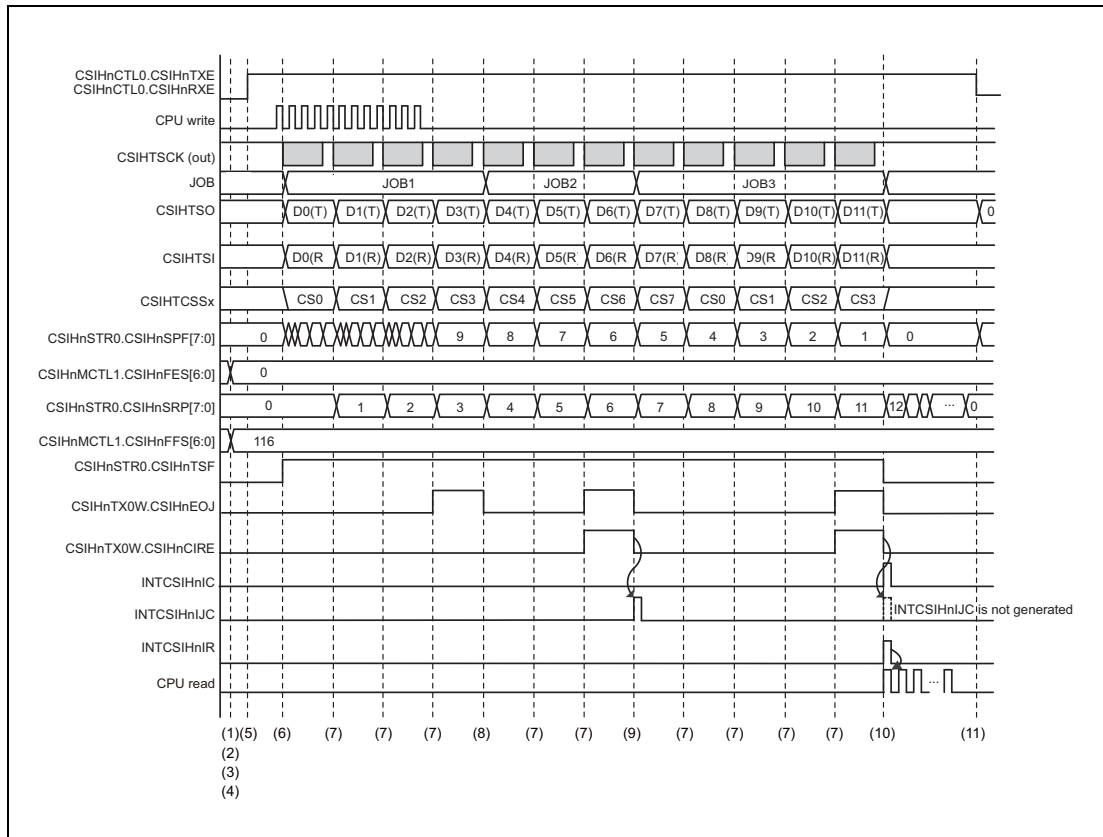


Figure 14.58 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Set communication protocol in the CSIHnCFGx register, set job mode disable and master mode by corresponding bits in the CSIHnCTL1 and CSIHnCTL2 registers, and specify FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] = 00_B. These examples use chip select signals CS0 to CS7.
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF is set to 0, CSIHnSTR0.CSIHnEMF is set to 1, and CSIHnSTR0.CSIHnSPF[7:0] is set to 00_H respectively.
4. With CSIHnMCTL1.CSIHnFES[6:0], specify the conditions for generating an interrupt request INTCSIHnIC. With CSIHnMCTL1.CSIHnFFS[6:0], specify the conditions for generating an interrupt request INTCSIHnIR.
5. Set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception) in the CSIHnCTL0 register. The CSIHnCTL0.CSIHnMBS bit must be cleared.
6. Write the transmission data to the transmission register CSIHnTX0W. When the transmission data becomes available, transmission starts automatically.
Make sure that CSIHnSTR0.CSIHnEMF is set to 0.
7. The current transmission is completed. The interrupt request, INTCSIHnIC is not generated because CSIHnFES[6:0] is not CSIHnSPF[7:0]
8. An interrupt request INTCSIHnIJC is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was transmitted with CSIHnTX0W.CSIHnCIRE = 0.
9. An interrupt request INTCSIHnIJC is generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was transmitted with CSIHnTX0W.CSIHnCIRE = 1.
10. Since CSIHnSTR0.CSIHnSPF[7:0] = CSIHnMCTL1.CSIHnFES[6:0], an interrupt request INTCSIHnIC is generated. Interrupt request INTCSIHnIJC is not generated because interrupt request INTCSIHnIC is generated instead. When CSIHnMCTL0.CSIHnFFS[6:0] becomes 128-CSIHnSTR0.CSIHnSRP[7:0], an interrupt request, INTCSIHnIR is generated. After the INTCSIHnIR interrupt is generated, the CPU starts reading the received data stored in the receive buffer.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize the power consumption of the CSIHn.

Section 15 Serial Communication Interface 3 (SCI3)

The serial communication interface 3 (SCI3: Serial Communication Interface 3) can handle two methods of serial communications: asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communication LSIs such as Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). Asynchronous mode is equipped with a serial communications function between multiple processors (multi-processor communications function).

15.1 Features of RH850/P1M-E SCI3

15.1.1 Number of Units and Channels

This microcontroller has the following number of SCI3 units.

Each unit has one channel SCI3. "Number of channels" is used with the same meaning as "number of units" in this section.

Table 15.1 Number of Units

| Product Name | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 3 | 3 |
| Name | SCI3n (n = 0 to 2) | SCI3n (n = 0 to 2) |

Table 15.2 SCI3 Unit Configurations and Channels

| Unit Name SCI3n | Channels per Unit | RH850/P1M-E 100 pins (3ch) | RH850/P1M-E 144 pins (3ch) |
|--------------------|----------------------|----------------------------------|----------------------------------|
| SCI30 | 1 | √ | √ |
| SCI31 | 1 | √ | √ |
| SCI32 | 1 | √ | √ |

Note: The channel name is the same as the unit name.

Table 15.3 Index

| Index | Meaning |
|-------|---|
| n | Throughout this section, the individual SCI3 units are identified by the index "n" (n = 0 to 2): for example, SCI3nSMR is the serial mode register. |

15.1.2 Register Base Address

SCI3 base addresses are listed in the following table.

SCI3 register addresses are given as offsets from these base addresses.

Table 15.4 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <SCI30_base> | FFDF 0000 _H |
| <SCI31_base> | FFDF 1000 _H |
| <SCI32_base> | FFDF 2000 _H |

15.1.3 Clock Supply

Clock supply by and to SCI3 is listed in the following table.

Table 15.5 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|-------------------------------------|
| SCI3n | PCLK | High-speed peripheral clock CLK_HSB |

15.1.4 Interrupt Request

SCI3 interrupt requests are listed in the following table.

Table 15.6 Interrupt Request

| Unit Interrupt Signal | Description | Interrupt Number | DMA/DTS Trigger Number |
|-----------------------|---------------------|------------------|------------------------|
| SCI30 | | | |
| INTSCI30ERI | Receive error | 106 | — |
| INTSCI30RXI | Receive data full | 107 | 87 |
| INTSCI30TXI | Transmit data empty | 108 | 88 |
| INTSCI30TEI | End of transmission | 109 | — |
| SCI31 | | | |
| INTSCI31ERI | Receive error | 110 | — |
| INTSCI31RXI | Receive data full | 111 | 89 |
| INTSCI31TXI | Transmit data empty | 112 | 90 |
| INTSCI31TEI | End of transmission | 113 | — |
| SCI32 | | | |
| INTSCI32ERI | Receive error | 218 | — |
| INTSCI32RXI | Receive data full | 219 | 91 |
| INTSCI32TXI | Transmit data empty | 220 | 92 |
| INTSCI32TEI | End of transmission | 221 | — |

15.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

15.1.6 External Input/Output Signals

External input/output signals of SCI3 are listed below.

Table 15.7 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal |
|------------------|-----|---------------------------|-----------------------------|
| SCI30 | | | |
| SCI0SCK | I | SCI30 serial clock input | SCI30SCI |
| | O | SCI30 serial clock output | SCI30SCO |
| SCI0RxD | I | SCI30 data input signal | SCI30RX |
| SCI0TxD | O | SCI30 data output signal | SCI30TX |
| SCI31 | | | |
| SCI1SCK | I | SCI31 serial clock input | SCI31SCI |
| | O | SCI31 serial clock output | SCI31SCO |
| SCI1RxD | I | SCI31 data input signal | SCI31RX |
| SCI1TxD | O | SCI31 data output signal | SCI31TX |
| SCI32 | | | |
| SCI2SCK | I | SCI32 serial clock input | SCI32SCI |
| | O | SCI32 serial clock output | SCI32SCO |
| SCI2RxD | I | SCI32 data input signal | SCI32RX |
| SCI2TxD | O | SCI32 data output signal | SCI32TX |

15.1.7 Combination of Pins and Ports

Combinations of SCI3 pins and ports are listed in the following table.

Table 15.8 Combinations of Pins and Ports

| Function | Pin Name | Port Name | | | | | |
|----------|------------------------|--------------------|-------------------------------|---------|---------|--------------------|---------|
| | | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 |
| SCI30 | SCI30RX | P0_2 | P2_5 | P3_7 | P4_5 | P5_0 | P5_0 |
| | SCI30TX | P0_3 ^{*1} | P2_6 | P3_12 | P4_6 | P5_1 | P5_1 |
| | SCI30SCI / SCI30SCO | P0_4 ^{*1} | P2_7 | P3_13 | P4_2 | P5_2 ^{*1} | P5_4 |
| SCI31 | SCI31RX | P2_8 | P5_5 | — | — | — | — |
| | SCI31TX | P2_9 | P5_6 | — | — | — | — |
| | SCI31SCI / SCI31SCO | P3_5 | P5_7 ^{*1} / P0_13 | — | — | — | — |
| SCI32 | SCI32RX | P3_4 | P5_8 ^{*1} / P0_2 | — | — | — | — |
| | SCI32TX | P3_3 | P5_9 | — | — | — | — |
| | SCI32SCI / SCI32SCO | P3_9 | P5_10 | — | — | — | — |

Note 1. Available in devices with 144-pin.

15.2 Outline of Functions

- The serial data communication mode can be configured for asynchronous communications or clock synchronous communications.
- Full-duplex communications are available. The independent transmitter unit and receiver unit allow simultaneous transmission and reception. Both the transmitter and the receiver have a double-buffered structure, enabling continuous data transmission and reception.
- An arbitrary bit rate is selectable with the on-chip baud rate generator. An external clock is also selectable as a transmission/reception clock source.
- LSB-first or MSB-first transfer is selectable (except for asynchronous 7-bit data.)
- Interrupt sources: 4 types consisting of transmit-end, transmit-data-empty, receive-data-full, and receive-error. Transmit-data-empty and receive-data-full interrupt sources can activate the DMAC.
- The bit rate modulation function can reduce errors averagely (even in a high bit rate) by correcting the output of the on-chip baud rate generator (except for the maximum speed in clock synchronous mode).
- The pin level of serial input data can be checked.
- On-chip 6-bit divider (PSC)

15.2.1 Serial Communication Modes

Asynchronous mode

- Data length: 7 bits or 8 bits selectable
- Stop bit length: 1 bit or 2 bits selectable
- Parity: Even parity, odd parity, or none selectable
- Receive error detection: Parity error, overrun error, and framing error
- Break detection: A break can be detected by reading a register when a framing error occurs

Clock synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun error

15.2.2 Block Diagram

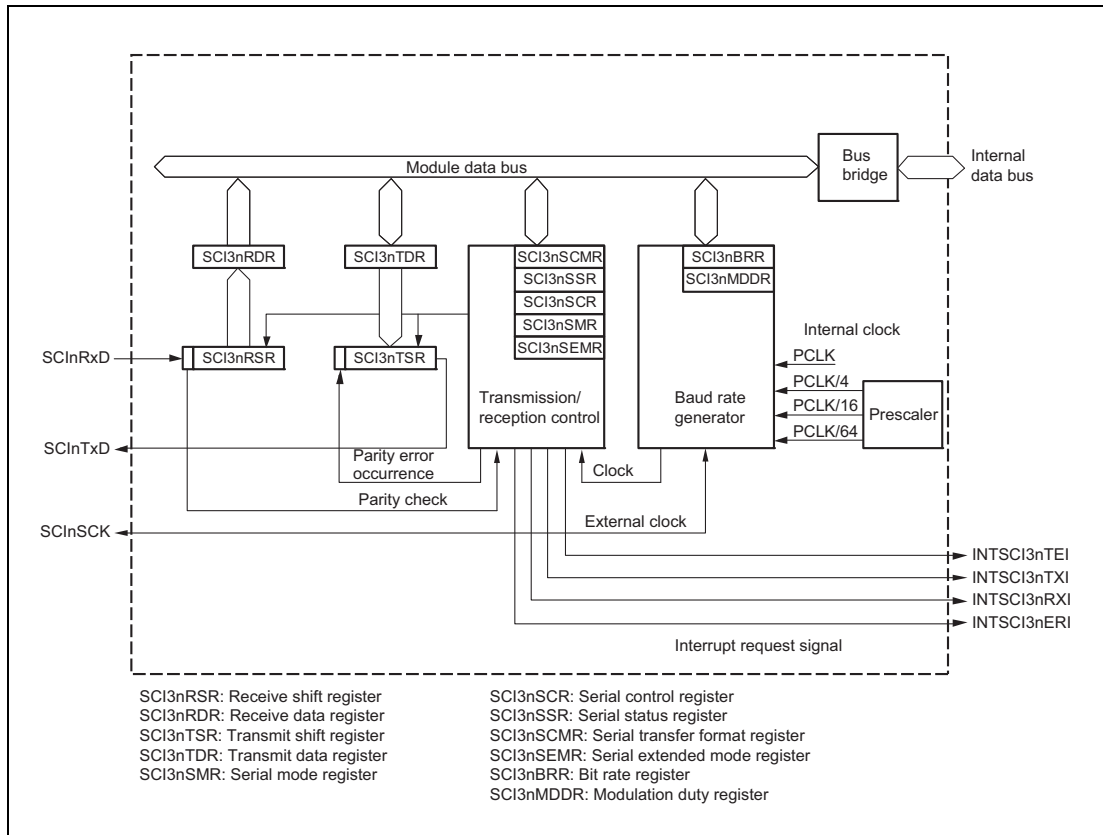


Figure 15.1 SCI3 Block Diagram

15.3 Register Descriptions

The SCI3 has the following registers. Some registers have limitations in read/write by the CPU.

For <SCI3n_base>, refer to **Section 15.1.2, Register Base Address**.

CAUTION

SCI3nBRR and SCI3nMDDR are allocated to the same address (relative address 4). The SCI3nMDDRS bit in SCI3nSEMR is used to switch these registers.

Table 15.9 Register Configuration

| Register Name | Symbol*1 | Value after Reset | Address | Access Size |
|---|------------------------|-------------------|----------------------------------|-------------|
| Receive shift register | SCI3nRSR | — | — | — |
| Serial mode register | SCI3nSMR | 00 _H | <SCI3n_base> + 0000 _H | 8 |
| Bit rate register / Modulation duty register | SCI3nBRR/ SCI3nMDDR | FF _H | <SCI3n_base> + 0004 _H | 8 |
| Serial control register | SCI3nSCR | 00 _H | <SCI3n_base> + 0008 _H | 8 |
| Transmit data register | SCI3nTDR | FF _H | <SCI3n_base> + 000C _H | 8 |
| Transmit shift register | SCI3nTSR | — | — | — |
| Serial status register | SCI3nSSR | 84 _H | <SCI3n_base> + 0010 _H | 8 |
| Receive data register | SCI3nRDR | 00 _H | <SCI3n_base> + 0014 _H | 8 |
| Serial transfer format register | SCI3nSCMR | F2 _H | <SCI3n_base> + 0018 _H | 8 |
| Serial extended mode register | SCI3nSEMR | 04 _H | <SCI3n_base> + 001C _H | 8 |

Note 1. n = 0 to 2

Relative addresses $4n + 1$, $4n + 2$, and $4n + 3$ ($n = 0$ to 2) are reserved areas. These areas are always read as 0. Writing is invalid.

15.3.1 SCI3nRSR — Receive Shift Register

SCI3nRSR is a shift register which is used to receive serial data input from the SCI3nRxD pin and convert it to parallel data. When one frame of data has been received, it is automatically transferred to SCI3nRDR. SCI3nRSR cannot be directly accessed by the CPU.

15.3.2 SCI3nRDR — Receive Data Register

SCI3nRDR is an 8-bit register to store receive data. The value of SCI3nRDR after a reset is 00_H. When one frame of data has been received, it is transferred from SCI3nRSR to this register allowing SCI3nRSR to receive the next data. SCI3nRSR and SCI3nRDR function as a double-buffer, allowing continuous receive operations. Be sure to check that the RDRF flag in SCI3nSSR is set to 1 before reading SCI3nRDR. SCI3nRDR cannot be written from the CPU.

When the data length is 7 bits, receive data is stored in bits 0 to 6 and bit 7 is fixed to 0 regardless of the SINV bit in SCI3nSCMR.

15.3.3 SCI3nTDR — Transmit Data Register

SCI3nTDR is an 8-bit register to store transmit data. The value of SCI3nTDR after a reset is FF_H. When SCI3nTSR empty is detected, the transmit data written to SCI3nTDR is transferred to SCI3nTSR and transmission starts. The double-buffered structure of SCI3nTDR and SCI3nTSR allows continuous serial transmission. If the next transmit data has been written to SCI3nTDR when one frame of data is sent, the transmit data is transferred to SCI3nTSR to continue transmission. SCI3nTDR can always be read and written by the CPU. Be sure to check that the TDRE flag in SCI3nSSR is set to 1 before writing transmit data to SCI3nTDR.

15.3.4 SCI3nTSR — Transmit Shift Register

SCI3nTSR is a shift register to transmit serial data. To perform serial data transmission, transmit data written to SCI3nTDR is automatically transferred to SCI3nTSR, and is then sent to the SCI3nTxD pin. SCI3nTSR cannot be directly accessed by the CPU.

15.3.5 SCI3nSMR — Serial Mode Register

SCI3nSMR is a register used to select the communication format and the clock source for the on-chip baud rate generator.

Access: This register can be read/written in 8-bit units.

Address: <SCI3n_base> + 0000_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | CM | CHR | PE | PM | STOP | MP | CKS1 | CKS0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 |

Note 1. Writable only when TE = RE = 0.

Table 15.10 SCI3nSMR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 | CM | Communication Mode 0: Asynchronous mode 1: Clock synchronous mode |
| 6 | CHR | Character Length (Valid only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. It is fixed to LSB-first and the MSB (bit 7) in SCI3nTDR is not sent in transmission. In clock synchronous mode, a data length of 8 bits is always used. |
| 5 | PE | Parity Enable (Valid only in asynchronous mode) When this bit is set to 1, a parity bit is added to transmit data and the parity bit is checked in reception. Regardless of the setting of this bit, no parity bit is added or checked in the multi-processor format. |
| 4 | PM | Parity Mode (Valid only when PE = 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity. When even parity is set, parity bit is added so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit is added so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd. |
| 3 | STOP | Stop Bit Length (Valid only in asynchronous mode) 0: 1 stop bit for transmission 1: 2 stop bits for transmission In reception, only the first stop bit is checked regardless of the setting of this bit. If the second stop bit is 0, it is treated as the start bit of the next transmission frame. |
| 2 | MP | Multi-Processor Mode (Valid only in asynchronous mode) When this bit is set to 1, the multi-processor communication function is enabled. In multi-processor mode, settings of the PE and PM bits are invalid. |
| 1, 0 | CKS[1:0] | Clock Select 1, 0 These bits select the clock source for the on-chip baud rate generator. 00: PCLK clock ($\alpha = 0$) 01: PCLK/4 clock ($\alpha = 1$) 10: PCLK/16 clock ($\alpha = 2$) 11: PCLK/64 clock ($\alpha = 3$) For the relation between the setting of these bits and the baud rate, see Section 15.3.10, SCI3nBRR — Bit Rate Register . The character α is the decimal notation of the value of α in Section 15.3.10, SCI3nBRR — Bit Rate Register . |

15.3.6 SCI3nSCR — Serial Control Register

SCI3nSCR is a register used for the transmission/reception control, interrupt control, and transmission/reception clock source selection listed below. For interrupt requests, see **Section 15.8, Interrupt Sources**.

Access: This register can be read/written in 8-bit units.

Address: <SCI3n_base> + 0008_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|-----|-------|-------|------|------|-------|-------|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W*1 | R/W*1 | R/W | R/W | R/W*2 | R/W*2 |

Note 1. While the CM bit in SCI3nSMR is 1, a value of 1 can be written only when TE = 0 and RE = 0. After the TE or RE bit is set to 1, only 0 can be written to TE and RE. While the CM bit in SCI3nSMR is 0, writing is enabled at any timing.

Note 2. Writable only when TE = 0 and RE = 0. Also, writable at the same time when TE = 0 and RE = 0 are written.

Table 15.11 SCI3nSCR Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | TIE | Transmit Interrupt Enable When this bit is set to 1, INTSCI3nTXI interrupt request is enabled. INTSCI3nTXI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0, or clearing the TIE bit. |
| 6 | RIE | Receive Interrupt Enable When this bit is set to 1, INTSCI3nRXI and INTSCI3nERI interrupt requests are enabled. INTSCI3nRXI and INTSCI3nERI interrupt requests can be cancelled by reading 1 from the RDRF, FER, PER, or ORER flag and then clearing the flag to 0, or clearing the RIE bit. |
| 5 | TE | Transmit Enable When this bit is set to 1, transmission is enabled. In this state, serial transmission is started by writing transmit data to SCI3nTDR and clearing the TDRE flag in SCI3nSSR to 0. Be sure to configure SCI3nSMR before setting the TE bit to 1 to determine the transmission format. When this bit is set to 0 to disable transmission, the TDRE flag in SCI3nSSR is fixed to 1. |
| 4 | RE | Receive Enable When this bit is set to 1, reception is enabled. In this state, serial reception is started by detecting a start bit in asynchronous mode or the input of synchronous clock in clock synchronous mode. Be sure to configure SCI3nSMR before setting the RE bit to 1 to determine the reception format. Even if reception is disabled by clearing this bit, the RDRF, FER, PER, or ORER flags are not affected and the previous value is retained. |
| 3 | MPIE | Multiprocessor Interrupt Enable (Valid only when MP in SCI3nSMR = 1 in asynchronous mode) When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the RDRF, FER, and ORER flags in SCI3nSSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, this bit is automatically cleared to 0, and normal reception is resumed. For details, see Section 15.5, Multi-Processor Communication Function . When the receive data includes MPB = 0 in SCI3nSSR, the receive data is not transferred from SCI3nRSR to SCI3nRDR, a receive error is not detected, and setting the RDRF, FER, and ORER flags in SCI3nSSR to 1 is disabled. When the receive data includes MPB = 1 in SCI3nSSR, the MPB bit in SCI3nSSR is set to 1, the MPIE bit is automatically cleared to 0, the INTSCI3nRXI and INTSCI3nERI interrupt requests are enabled (if the RIE bit in SCI3nSCR is set to 1), and setting the FER and ORER flags to 1 is enabled. |

Table 15.11 SCI3nSCR Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 2 | TEIE | Transmit End Interrupt Enable When this bit is set to 1, INTSCI3nTEI interrupt request is enabled. INTSCI3nTEI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0 to clear the TEND flag to 0, or clearing the TEIE bit. |
| 1, 0 | CKE[1:0] | Clock Enable 1, 0 These bits select the clock source and the SCInSCK pin function. For asynchronous mode 00: On-chip baud rate generator (The SCInSCK pin functions as an input/output port.) 01: On-chip baud rate generator (The clock with the same frequency as the bit rate is output from the SCInSCK pin.) 1X: Setting prohibited For clock synchronous mode 0X: Internal clock (The SCInSCK pin functions as a clock output pin.) 1X: External clock (The SCInSCK pin functions as a clock input pin.) |

Note: X: Don't care

15.3.7 SCI3nSSR — Serial Status Register

SCI3nSSR consists of SCI3 status flags and multi-processor transmit/receive bits.
The TDRE, RDRF, ORER, PER, and FER flags can be cleared only.

Access: This register can be read/written in 8-bit units.

Address: <SCI3n_base> + 0010_H

Value after reset: 84_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------|---------|---------|---------|---------|------|-----|------|
| | TDRE | RDRF | ORER | FER | PER | TEND | MPB | MPBT |
| Value after reset | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R | R | R/W |

Note 1. Only 0 can be written to clear the flag.

Table 15.12 SCI3nSSR Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 | TDRE | Transmit Data Register Empty Indicates whether or not transmit data exists in SCI3nTDR. [Setting conditions] <ul style="list-style-type: none"> When the TE bit in SCI3nSCR is 0 When the data in SCI3nTDR has been transferred to SCI3nTSR so that new data can be written to SCI3nTDR [Clearing condition] <ul style="list-style-type: none"> Writing 0 to TDRE after reading TDRE = 1 When transmit data is written to SCI3nTDR while TE = 1 |
| 6 | RDRF | Receive Data Register Full Indicates whether or not receive data exists in SCI3nRDR [Setting condition] <ul style="list-style-type: none"> When reception finishes successfully and the receive data is transferred from SCI3nRSR to SCI3nRDR [Clearing conditions] <ul style="list-style-type: none"> Writing 0 to RDRF after reading RDRF = 1 When data is read from SCI3nRDR Even when the RE bit in SCI3nSCR is cleared, the RDRF flag is not affected and the previous value is retained. Note that completing the reception of the next data with the RDRF flag set to 1 will cause an overrun error, resulting in the loss of the receive data. |
| 5 | ORER | Overrun Error Indicates that an overrun error has occurred during reception and the reception abended. [Setting condition] <ul style="list-style-type: none"> When the next data is received while RDRF = 1 In SCI3nRDR, data received prior to the overrun error is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be continued. In clock synchronous mode, serial transmission also cannot be continued. [Clearing condition] <ul style="list-style-type: none"> Writing 0 to ORER after reading ORER = 1 Even when the RE bit in SCI3nSCR is cleared, the ORER flag is not affected and the previous value is retained. |

Table 15.12 SCI3nSSR Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 4 | FER | <p>Framing Error</p> <p>Indicates that a framing error has occurred during reception in asynchronous mode and the reception abended.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> When the stop bit is 0 In 2-stop-bit mode, only whether the first stop bit is 1 is checked but the second stop bit is not checked. Although the receive data that existed when a framing error has occurred is transferred to SCI3nRDR, the RDRF flag is not set. In addition, while the FER flag is set to 1, the subsequent receive data is not transferred to SCI3nRDR. <p>[Clearing condition]</p> <ul style="list-style-type: none"> Writing 0 to FER after reading FER = 1 Even when the RE bit in SCI3nSCR is cleared to 0, the FER flag is not affected and the previous value is retained. |
| 3 | PER | <p>Parity Error</p> <p>Indicates that a parity error has occurred during reception in asynchronous mode and the reception abended.</p> <p>[Setting condition]</p> <ul style="list-style-type: none"> Although the receive data that existed when a parity error has occurred is transferred to SCI3nRDR, the RDRF flag is not set. In addition, while the PER flag is set to 1, the subsequent receive data is not transferred to SCI3nRDR. <p>[Clearing condition]</p> <ul style="list-style-type: none"> Writing 0 to PER after reading PER = 1 Even when the RE bit in SCI3nSCR is cleared to 0, the PER flag is not affected and the previous value is retained. |
| 2 | TEND | <p>Transmit End</p> <p>Indicates that a transmission is completed.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> When the TE bit in SCI3nSCR is 0 When the TDRE flag is 1 while the last bit of a transmit character is being transmitted <p>[Clearing condition]</p> <ul style="list-style-type: none"> Writing 0 to the TDRE flag after reading TDRE = 1 When transmit data is written to SCI3nTDR while TE = 1 |
| 1 | MPB | <p>Multi-processor Bit</p> <p>Holds the value of the multi-processor bit in the receive frame.</p> |
| 0 | MPBT | <p>Multi-processor Bit Transfer</p> <p>Sets the value of the multi-processor bit to be added to the transmit frame.</p> |

15.3.8 SCI3nSCMR — Serial Transfer Format Register

SCI3nSCMR is a register to select the communication format which can be commonly configured for both asynchronous mode and clock synchronous mode.

Access: This register can be read/written in 8-bit units.

Address: <SCI3n_base> + 0018_H

Value after reset: F2_H

| | | | | | | | | |
|-------------------|---|---|---|---|-------|-------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | SDIR | SINV | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R/W*1 | R/W*1 | R | R |

Note 1. Writable only when TE = 0 and RE = 0.

Table 15.13 SCI3nSCMR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | SDIR | Serial Data Transfer Direction (Valid in asynchronous mode and clock synchronous mode) This bit is used to select the direction of serial/parallel conversion. 0: Transfer with LSB-first 1: Transfer with MSB-first This bit is valid only when the transfer format is 8-bit data. For 7-bit data, LSB-first transfer is used. |
| 2 | SINV | Serial Data Invert (Valid in asynchronous mode and clock synchronous mode) This bit is used to invert the transmit/receive data logic level. The SINV bit does not affect the logic level of the start bit, stop bit, parity bit, and multi-processor bit. To invert the parity bit, invert the PM bit in SCI3nSMR. 0: SCI3nTDR data is transmitted as it is, and receive data is stored in SCI3nRDR as it is. 1: SCI3nTDR data is inverted and transmitted, and receive data is inverted and stored in SCI3nRDR. |
| 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

15.3.9 SCI3nSEMR — Serial Extended Mode Register

SCI3nSEMR is a register to select a 1-bit period.

Access: This register can be read/written in 8-bit units.

Address: <SCI3n_base> + 001C_H

Value after reset: 04_H

| | | | | | | | | |
|-------------------|-------|-------|---|---|-------|--------|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | BRME | MDDRS | — | — | ABCS | RXDMON | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R/W*1 | R/W*1 | R | R | R/W*1 | R | R | R |

Note 1. Writable only when TE = 0 and RE = 0.

Table 15.14 SCI3nSEMR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | BRME | Bit Rate Modulation Enable When this bit is set to 1, the bit rate modulation function is enabled. |
| 6 | MDDRS | Modulation Duty Register Select This bit is used to select an accessible register. 0: SCI3nBRR is accessible. 1: SCI3nMDDR is accessible. |
| 5, 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | ABCS | Asynchronous Reference Clock Select (Valid only in asynchronous mode) This bit is used to select the reference clock for 1-bit period. 0: Operates on the reference clock with a frequency of 16 times the transfer rate. 1: Operates on the reference clock with a frequency of 8 times the transfer rate (double-speed operation). |
| 2 | RXDMON | Serial Input Data Monitor This bit indicates the SCInRxD pin state. 0: SCInRxD pin state is low level. 1: SCInRxD pin state is high level. |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

15.3.10 SCI3nBRR — Bit Rate Register

SCI3nBRR is an 8-bit register to adjust the bit rate. Since each SCI3 channel has an independent baud rate generator, different bit rates can be set for each channel. **Table 15.16** shows the relationship between the setting (N) in SCI3nBRR and the bit rate (B) in normal asynchronous mode and clock synchronous mode. The value of SCI3nBRR after reset is FF_H. SCI3nBRR is allocated at the same address as SCI3nMDDR and is selected when MDDRS in SCI3nSEMR is 0. This register is writable only when TE = 0 and RE = 0.

Access: This register can be read/written in 8-bit units.

Address: <SCI3n_base> + 0004_H

Value after reset: FF_H

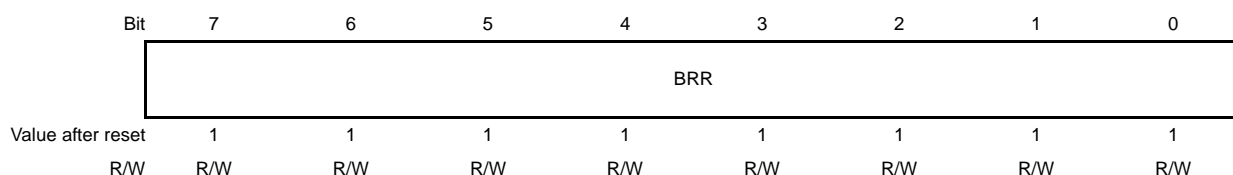


Table 15.15 SCI3nBRR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 0 | BRR | Baud rate generator setting (0 ≤ N ≤ 255) |

Table 15.16 Relationship between Setting N in SCI3nBRR and Bit Rate B

| Mode | ABCS Setting | Bit Rate | Mean Error |
|-------------------|--------------|---|--|
| Asynchronous | 0 | $B = \frac{PCLK \times 10^6}{64 \times 2^{2\alpha-1} \times (N+1)}$ | $Error(\%) = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2\alpha-1} \times (N+1)} - 1 \right\} \times 100$ |
| | 1 | $B = \frac{PCLK \times 10^6}{32 \times 2^{2\alpha-1} \times (N+1)}$ | $Error(\%) = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2\alpha-1} \times (N+1)} - 1 \right\} \times 100$ |
| Clock synchronous | — | $B = \frac{PCLK \times 10^6}{8 \times 2^{2\alpha-1} \times (N+1)}$ | |

Note: B: Bit rate (bps)
 N: SCI3nBRR setting value for baud rate generator (0 ≤ N ≤ 255)
 PCLK: Operating frequency (MHz)
 α: Determined by the SCI3nSMR setting value as shown in the following table.

| SCI3nSMR Setting Value | | α |
|------------------------|------|---|
| CKS1 | CKS0 | |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

Table 15.17 lists sample N settings of the SCI3nBRR register in asynchronous mode. **Table 15.18** shows the maximum configurable bit rates.

Table 15.17 Examples of BRR Settings for Bit Rates (Asynchronous Mode) (80 MHz)

| PCLK (MHz) | α | N | Baud Rate | |
|------------|----------|-----|--------------------|--------------------|
| | | | SCI3nSEMR.ABCS = 0 | SCI3nSEMR.ABCS = 1 |
| 80 | 0 | 0 | 2.500 MHz | 5.000 MHz |
| 80 | 0 | 1 | 1.250 MHz | 2.500 MHz |
| 80 | 0 | 2 | 0.833 MHz | 1.667 MHz |
| 80 | 0 | 3 | 0.625 MHz | 1.250 MHz |
| 80 | 0 | 4 | 0.500 MHz | 1.000 MHz |
| 80 | 0 | 5 | 0.417 MHz | 0.833 MHz |
| 80 | 0 | 6 | 0.357 MHz | 0.714 MHz |
| 80 | 0 | 7 | 0.313 MHz | 0.625 MHz |
| : | | | | |
| 80 | 1 | 0 | 0.625 MHz | 1.250 MHz |
| 80 | 1 | 1 | 0.313 MHz | 0.625 MHz |
| 80 | 1 | 2 | 0.208 MHz | 0.417 MHz |
| 80 | 1 | 3 | 0.156 MHz | 0.313 MHz |
| 80 | 1 | 4 | 0.125 MHz | 0.250 MHz |
| 80 | 1 | 5 | 0.104 MHz | 0.208 MHz |
| 80 | 1 | 6 | 0.089 MHz | 0.179 MHz |
| 80 | 1 | 7 | 0.078 MHz | 0.156 MHz |
| : | | | | |
| 80 | 3 | 242 | 160.751 Hz | 321.502 Hz |
| 80 | 3 | 243 | 160.092 Hz | 320.184 Hz |
| 80 | 3 | 244 | 159.439 Hz | 318.878 Hz |
| 80 | 3 | 245 | 158.791 Hz | 317.581 Hz |
| 80 | 3 | 246 | 158.148 Hz | 316.296 Hz |
| 80 | 3 | 247 | 157.510 Hz | 315.020 Hz |
| 80 | 3 | 248 | 156.878 Hz | 313.755 Hz |
| 80 | 3 | 249 | 156.250 Hz | 312.500 Hz |
| 80 | 3 | 250 | 155.627 Hz | 311.255 Hz |
| 80 | 3 | 251 | 155.010 Hz | 310.020 Hz |
| 80 | 3 | 252 | 154.397 Hz | 308.794 Hz |
| 80 | 3 | 253 | 153.789 Hz | 307.579 Hz |
| 80 | 3 | 254 | 153.186 Hz | 306.373 Hz |
| 80 | 3 | 255 | 152.588 Hz | 305.176 Hz |

Table 15.18 Maximum Bit Rate (Asynchronous Mode)

| PCLK (MHz) | Setting | | | Maximum Serial Clock Frequency |
|------------|--------------|----------|---|--------------------------------|
| | ABCS Setting | α | N | |
| 80 | 1 | 0 | 0 | 5 MHz |

Table 15.19 lists sample N settings of the SCI3nBRR register in clock synchronous mode.

Table 15.20 shows the maximum configurable bit rates.

Table 15.19 Examples of Bit Rate Settings for Clock Synchronous Mode (Master Mode)

| α | N | Baud Rate |
|----------|-----|--------------------|
| | | PCLK = 80 MHz |
| 0 | 0 | Setting prohibited |
| 0 | 1 | Setting prohibited |
| 0 | 2 | Setting prohibited |
| 0 | 3 | 5.000 MHz |
| 0 | 4 | 4.000 MHz |
| 0 | 5 | 3.333 MHz |
| 0 | 6 | 2.857 MHz |
| 0 | 7 | 2.500 MHz |
| : | | |
| 1 | 0 | 5.000 MHz |
| 1 | 1 | 2.500 MHz |
| 1 | 2 | 1.667 MHz |
| 1 | 3 | 1.250 MHz |
| 1 | 4 | 1.000 MHz |
| 1 | 5 | 0.833 MHz |
| 1 | 6 | 0.714 MHz |
| 1 | 7 | 0.625 MHz |
| : | | |
| 3 | 242 | 1286.008 Hz |
| 3 | 243 | 1280.738 Hz |
| 3 | 244 | 1275.510 Hz |
| 3 | 245 | 1270.325 Hz |
| 3 | 246 | 1265.182 Hz |
| 3 | 247 | 1260.081 Hz |
| 3 | 248 | 1255.020 Hz |
| 3 | 249 | 1250.000 Hz |
| 3 | 250 | 1245.020 Hz |
| 3 | 251 | 1240.079 Hz |
| 3 | 252 | 1235.178 Hz |
| 3 | 253 | 1230.315 Hz |
| 3 | 254 | 1225.490 Hz |
| 3 | 255 | 1220.703 Hz |

Table 15.20 Maximum Bit Rate (Clock Synchronous Mode) (Master Mode)

| PCLK (MHz) | α | N | Maximum Serial Clock Frequency |
|------------|----------|---|--------------------------------|
| 80 | 0 | 3 | 5 MHz |

15.3.11 SCI3nMDDR — Modulation Duty Register

SCI3nMDDR is a register to correct the bit rate adjusted by SCI3nBRR. The value of SCI3nMDDR after a reset is FF_H. When the BRME bit in SCI3nSEMR is set to 1, the bit rate generated by the on-chip baud rate generator is corrected to SCI3nMDDR/256 on average. **Table 15.22** shows the relationship between the SCI3nMDDR setting and the bit rate B. SCI3nMDDR is allocated at the same address as SCI3nBRR and is selected when MDDRS in SCI3nSEMR is 1. This register is writable only when TE = 0 and RE = 0. Bit 7 is fixed to 1.

Access: This register can be read/written in 8-bit units.

Address: <SCI3n_base> + 0004_H

Value after reset: FF_H

| | | | | | | | | |
|-------------------|------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MDDR | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 15.21 SCI3nMDDR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 0 | MDDR | Baud rate generator setting value (128 ≤ MDDR ≤ 255) |

Table 15.22 Relationship between SCI3nMDDR Setting Value and Bit Rate B when Bit Rate Modulation Function Is Used

| Mode | ABCS Setting | Bit Rate | Mean Error |
|-------------------|--------------|---|---|
| Asynchronous | 0 | $B = \frac{PCLK \times 10^6}{64 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)}$ | $\text{Error(\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)} - 1 \right\} \times 100$ |
| | 1 | $B = \frac{PCLK \times 10^6}{32 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)}$ | $\text{Error(\%)} = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)} - 1 \right\} \times 100$ |
| Clock synchronous | — | $B = \frac{PCLK \times 10^6}{8 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)}$ | |

Note: B: Bit rate (bps)
 N: SCI3nBRR setting value of baud rate generator (0 ≤ N ≤ 255)
 PCLK: Operating frequency (MHz)
 α: See **Table 15.16, Relationship between Setting N in SCI3nBRR and Bit Rate B.**
 SCI3nMDDR: SCI3nMDDR setting (128 ≤ SCI3nMDDR ≤ 255)

15.4 Operation in Asynchronous Mode

Figure 15.2 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communication line is usually held in the mark state (high level). The SCI3 monitors the communication line, and when the SCI3 detects the space state (low level), it recognizes a start bit and starts serial communications. Inside the SCI3, the transmitter and the receiver are independent, enabling full-duplex communications. Both the transmitter and the receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

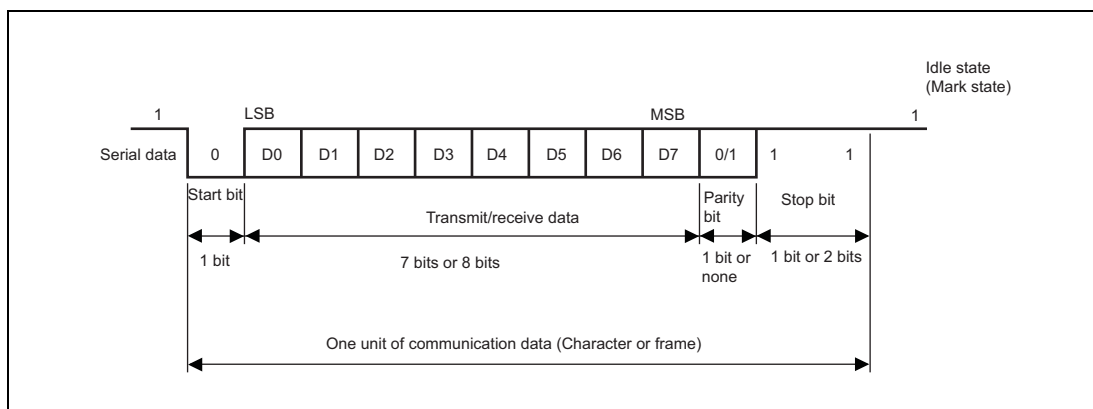


Figure 15.2 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, Two Stop Bits)

15.4.1 Transmission/Reception Format

Table 15.23 lists the transmission/reception formats that can be configured in asynchronous mode. Any of 12 transmission/reception formats can be selected according to the SCI3nSMR setting. For details of the multi-processor bit, see **Section 15.5, Multi-Processor Communication Function**.

Table 15.23 Serial Transmission/Reception Formats (Asynchronous Mode)

| SMR Setting | | | | Serial Transmission/Reception Format and Frame Length | | | | | | | | | | | | |
|-------------|----|----|------|---|------------|---|---|---|---|---|---|------|------|------|------|--|
| CHR | PE | MP | STOP | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
| 0 | 0 | 0 | 0 | S | 8-bit data | | | | | | | | STOP | | | |
| 0 | 0 | 0 | 1 | S | 8-bit data | | | | | | | | STOP | STOP | | |
| 0 | 1 | 0 | 0 | S | 8-bit data | | | | | | | | P | STOP | | |
| 0 | 1 | 0 | 1 | S | 8-bit data | | | | | | | | P | STOP | STOP | |
| 1 | 0 | 0 | 0 | S | 7-bit data | | | | | | | STOP | | | | |
| 1 | 0 | 0 | 1 | S | 7-bit data | | | | | | | STOP | STOP | | | |
| 1 | 1 | 0 | 0 | S | 7-bit data | | | | | | | P | STOP | | | |
| 1 | 1 | 0 | 1 | S | 7-bit data | | | | | | | P | STOP | STOP | | |
| 0 | — | 1 | 0 | S | 8-bit data | | | | | | | | MPB | STOP | | |
| 0 | — | 1 | 1 | S | 8-bit data | | | | | | | | MPB | STOP | STOP | |
| 1 | — | 1 | 0 | S | 7-bit data | | | | | | | MPB | STOP | | | |
| 1 | — | 1 | 1 | S | 7-bit data | | | | | | | MPB | STOP | STOP | | |

Note: S: Start bit
 STOP: Stop bit
 P: Parity bit
 MPB: Multi-processor bit

15.4.2 Receive Data Sampling Timing and Reception Margin

In asynchronous mode, the SCI3 operates on a reference clock with a frequency of 16 times (8 times for the double-speed mode) the bit rate. In reception, the SCI3 samples the falling edge of the beginning of the start bit (low level) using the reference clock and performs internal synchronization. As shown in **Figure 15.3**, data is latched at the middle of each bit by sampling receive data at the rising edge of the eighth pulse (fourth pulse for the double-speed mode) of the reference clock.

Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100[\%] \dots \text{formula(1)}$$

- M: Reception margin
- N: Ratio of bit rate to clock (N = 16 when ABCS in SCI3nSEMR = 0, N = 8 when ABCS = 1)
- D: Duty cycle of clock (D = 0.5 to 1.0)
- L: Frame length (L = 9 to 12)
- F: Absolute value of clock frequency deviation

Assuming that F (absolute value of clock frequency deviation) = 0, D (duty cycle of clock) = 0.5, and N = 16 in formula (1), the reception margin is obtained by the formula below.

$$M = \left\{ 0.5 - \frac{1}{(2 \times 16)} \right\} \times 100[\%] = 46.875\%$$

However, this is only the calculated value, and a margin of 20% to 30% should be allowed in system design.

When the bit rate modulation function is used, the reference clock frequency is corrected on average.

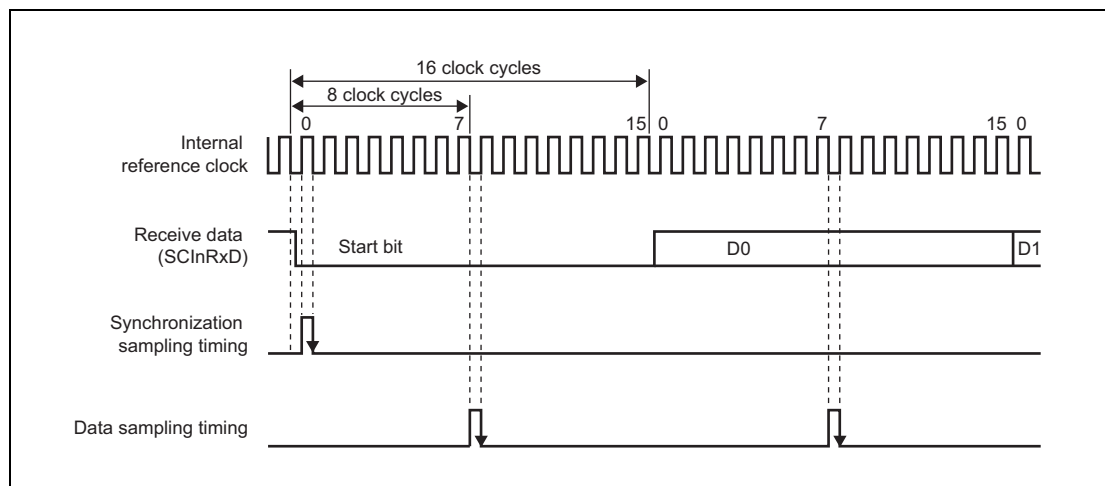


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

15.4.3 Clock

An internal clock generated by the on-chip baud rate generator can be selected as the SCI3's transmission/reception clock according to the settings of the CM bit in SCI3nSMR and the CKE1 and CKE0 bits in SCI3nSCR. When the SCI3 is operated on an internal clock, the clock can be output from the SCInSCK pin.

For details of clock synchronous mode, see **Section 15.6, Operation in Clock Synchronous Mode**.

In asynchronous mode, the frequency of the clock output is equal to the bit rate and the phase is such that the rising edge of the clock comes at the center of the transmit data, as shown in **Figure 15.4**.

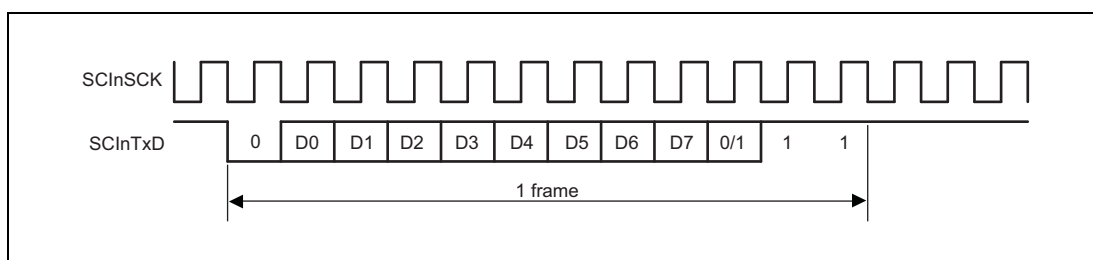


Figure 15.4 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode)

15.4.4 Double-Speed Operation

In addition to the operation described in **Section 15.4.3, Clock**, double-speed operation is enabled by the setting of the ABCS bit in SCI3nSEMR.

In double-speed operation, the same operation on a clock with a frequency of 16 times the bit rate in normal operation can be conducted on a clock with a frequency of 8 times the bit rate, meaning that the SCI3 operates on a double transfer rate using the same reference clock.

15.4.5 SCI3 Initialization (Asynchronous Mode)

Before transmitting and receiving data, clear the TE and RE bits in SCI3nSCR and then initialize the SCI3 according to the sample flowchart in **Figure 15.5**. Before changing the operating mode or transfer format, be sure to clear the TE and RE bits to 0. Note that clearing the TE bit to 0 sets the TDRE flag to 1, but clearing the RE bit to 0 initializes neither the RDRF, PER, FER, and ORER flags nor SCI3nRDR.

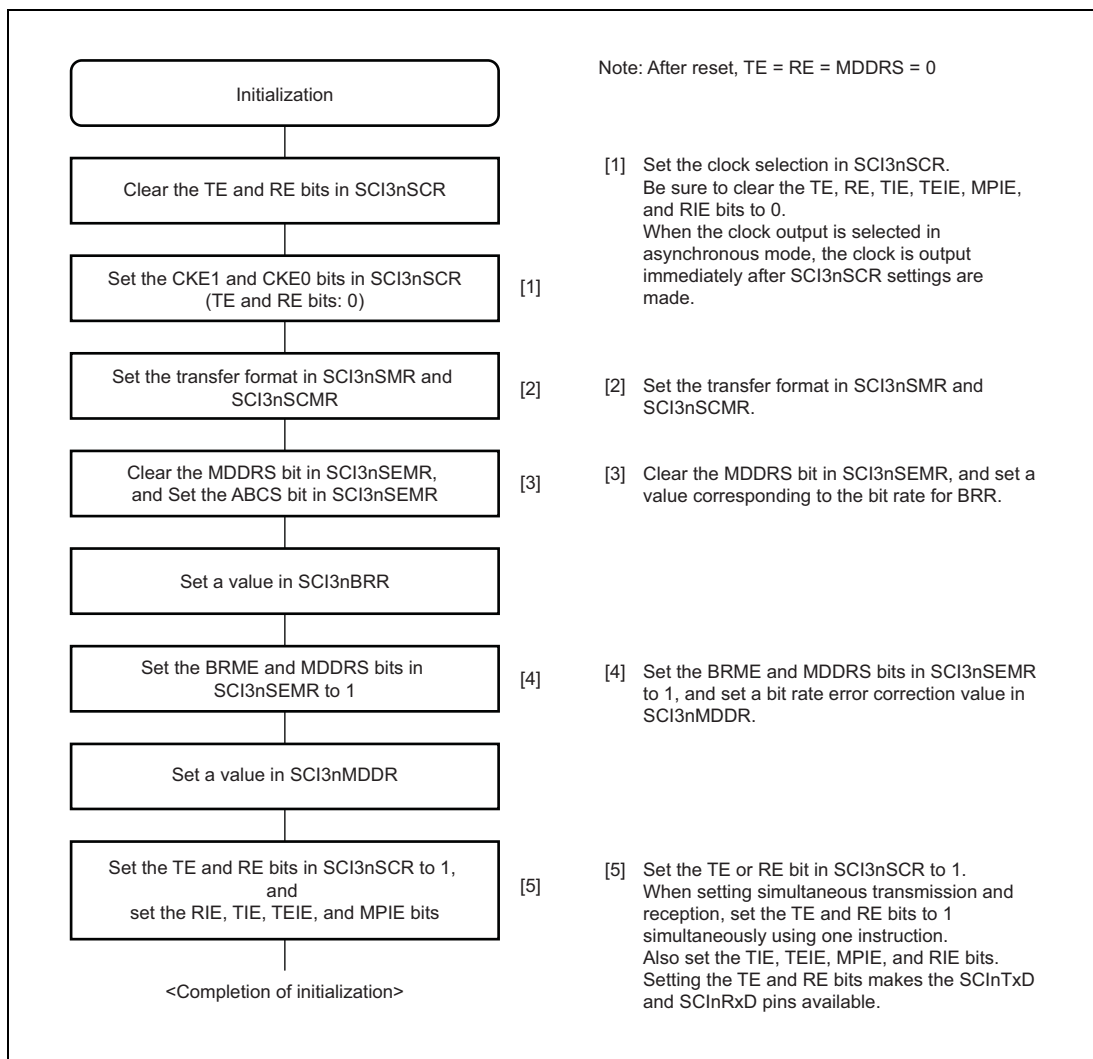


Figure 15.5 Sample Flowchart for SCI3 Initialization

15.4.6 Serial Data Transmission (Asynchronous Mode)

Figure 15.6 shows an example of operation for data transmission in asynchronous mode. In data transmission, the SCI3 operates as described below.

1. When transmit data is written to SCI3nTDR, the TDRE flag is automatically cleared to 0. The SCI3 monitors the TDRE flag in SCI3nSSR. When the flag is cleared, the SCI3 recognizes that data has been written to SCI3nTDR and transfers data from SCI3nTDR to SCI3nTSR. When writing transmit data to SCI3nTDR at a trigger of INTSCI3nTXI interrupt request, set the TIE bit to 1 and then set the TE bit to 1 or set both TIE and TE bits simultaneously with one instruction to generate a INTSCI3nTXI interrupt request for starting data transfer.
2. Transmission starts after data is transferred from SCI3nTDR to SCI3nTSR and the TDRE flag is set to 1. If the TIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to SCI3nTDR in this INTSCI3nTXI interrupt processing routine before transmission of the previously transferred data is completed. When an INTSCI3nTEI interrupt request is used, the TIE bit is cleared to 0 after the last transmit data has been written to SCI3nTDR, and the TEIE bit is set to 1.
3. Data is sent from the SCInTxD pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The TDRE flag is checked when the stop bit is output.
5. When the TDRE flag is 0, the next transmit data is transferred from SCI3nTDR to SCI3nTSR. After the stop bit has been sent, transmission of the next frame starts.
6. When the TDRE flag is 1, the TEND flag in SCI3nSSR is set to 1, the stop bit is sent, and then 1 is output to enter the mark state. If the TEIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTEI interrupt request is generated.

Figure 15.7 shows a sample flowchart for data transmission. **Figure 15.8** shows a sample flowchart for stopping the SCI3 after data transmission.

Note about the operation when transmission is enabled in asynchronous mode:

When the setting of the TE bit is changed from 0 to 1, a high level (preamble) is output for one frame. If transmit data is written to SCI3nTDR while the preamble is being output, that data will be transferred from SCI3nTDR to SCI3nTSR after preamble output is complete.

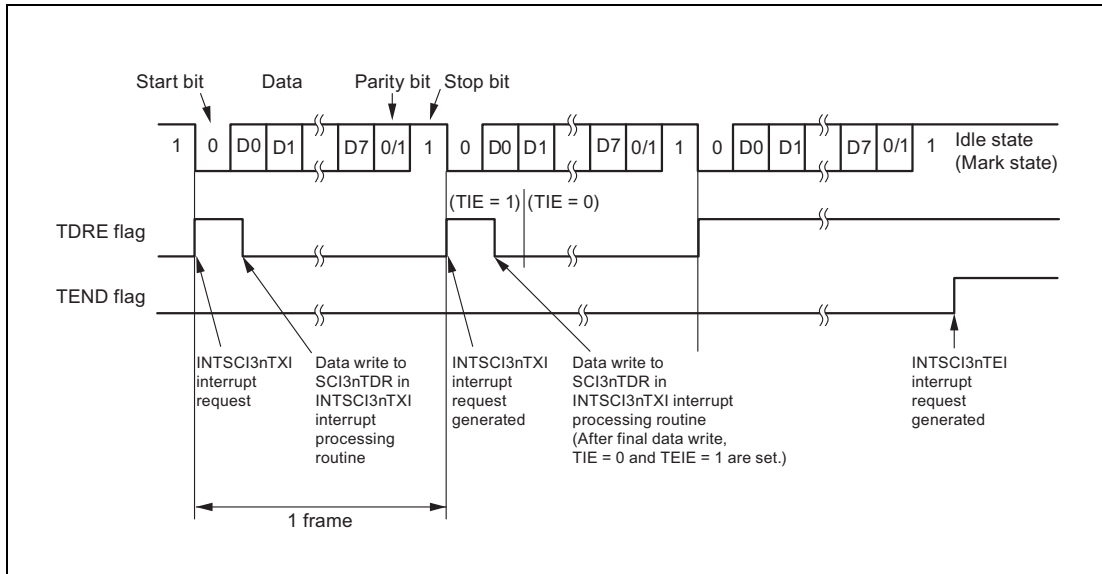


Figure 15.6 Example of Operation for Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

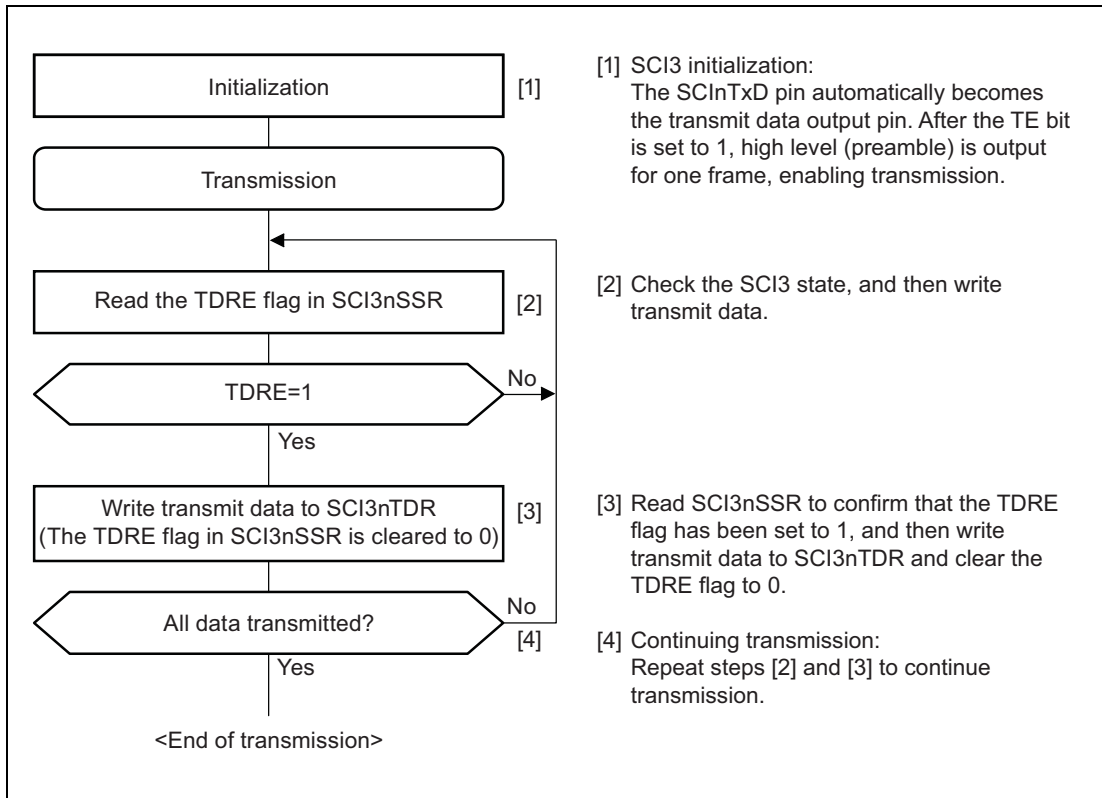


Figure 15.7 Example of Serial Transmission Flowchart

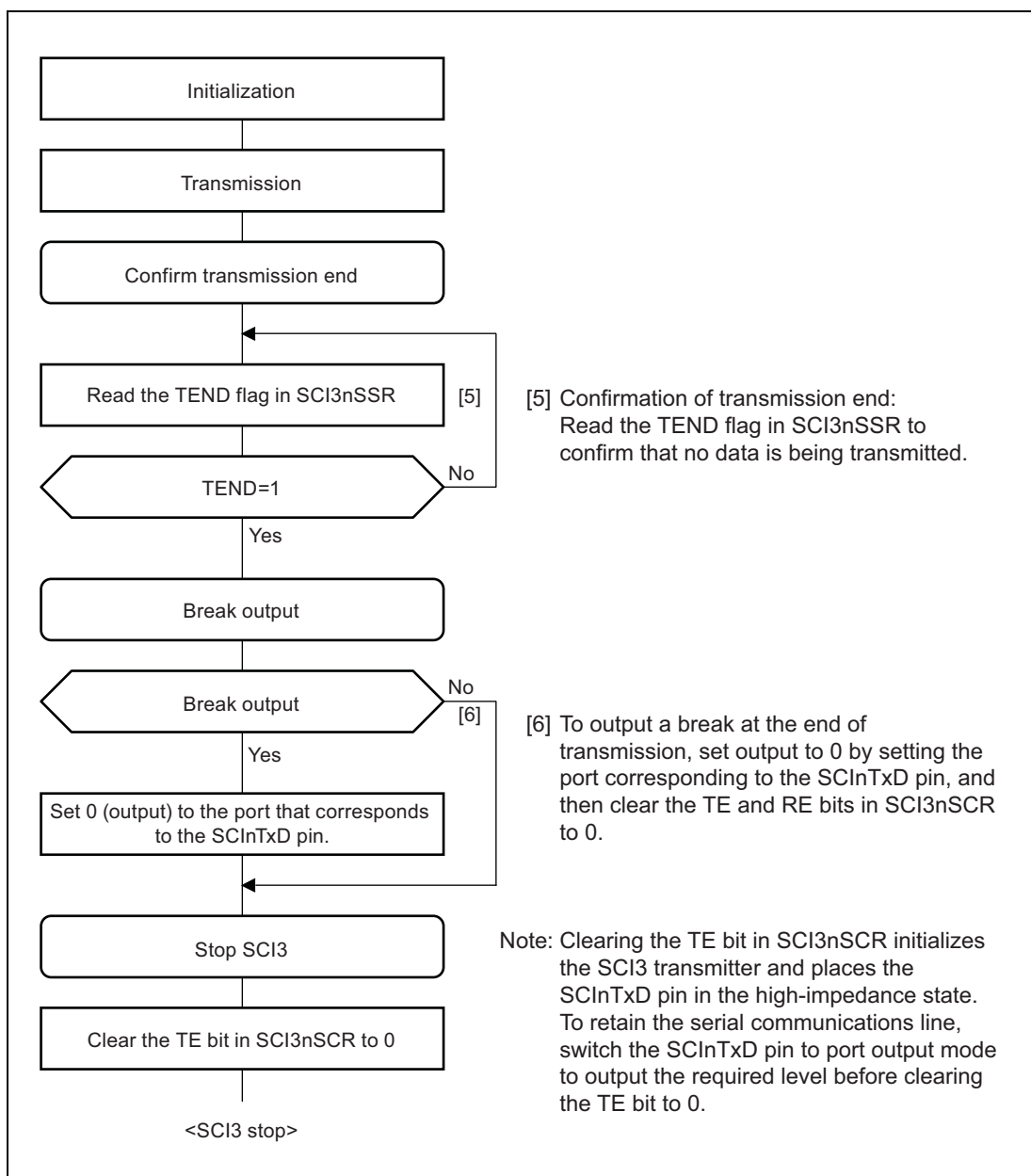


Figure 15.8 Example Flowchart for Stopping the SCI3 after Serial Transmission

15.4.7 Serial Data Reception (Asynchronous Mode)

Figure 15.9 shows an example of the operation for data reception in asynchronous mode. In data reception, the SCI3 operates as described below.

1. SCI3 monitors the communications line and upon detection of a start bit, it performs internal synchronization, stores receive data in SCI3nRSR, and checks the parity bit and the stop bit.
2. When an overrun error occurs (the next data has been received with the RDRF flag in SCI3nSSR set to 1), the ORER flag in SCI3nSSR is set to 1. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated. Receive data is not transferred to SCI3nRDR. The RDRF flag retains the state of being set to 1.
3. When a parity error is detected, the PER flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated.
4. When a framing error (when the stop bit is 0) is detected, the FER flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated.
5. When reception finishes successfully, the RDRF flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nRXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to SCI3nRDR in this INTSCI3nRXI interrupt processing routine before reception of the next receive data is completed. Reading SCI3nRDR automatically clears the RDRF flag to 0.

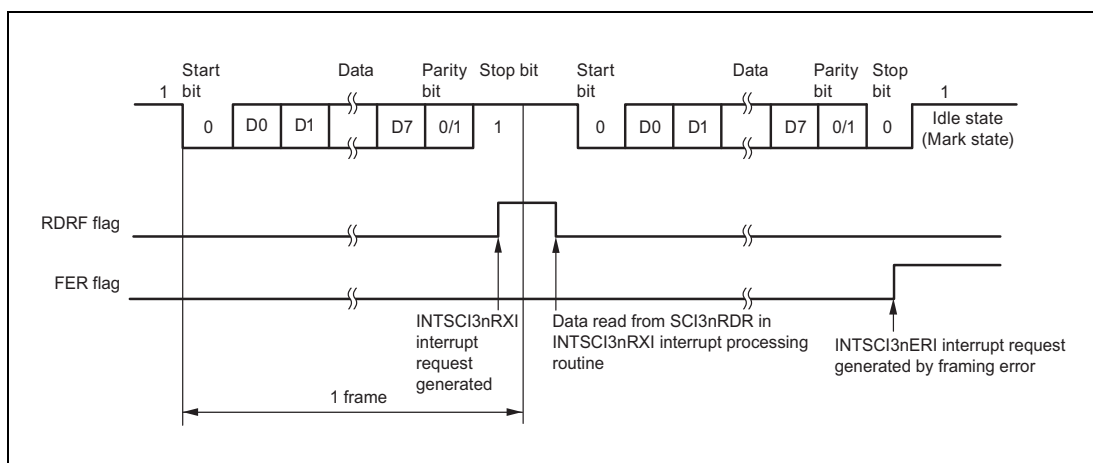


Figure 15.9 Example of Operation for Reception in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Table 15.24 lists the states of the SCI3nSSR status flags and receive data handling when a receive error is detected. When a receive error is detected, the RDRF flag retains the status before receiving the data. Subsequent data reception is disabled while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF flags before continuing data reception. **Figure 15.10** shows a sample flowchart for data reception.

Table 15.24 SCI3nSSR Status Flags and Receive Data Handling

| SCI3nSSR Status Flags | | | | Receive Data | Receive Status |
|-----------------------|------|-----|-----|-------------------------|--|
| RDRF ^{*1} | ORER | FER | PER | | |
| 1 | 0 | 0 | 0 | Transferred to SCI3nRDR | Successful reception |
| 0 | 0 | 1 | 0 | Transferred to SCI3nRDR | Framing error |
| 0 | 0 | 0 | 1 | Transferred to SCI3nRDR | Parity error |
| 0 | 0 | 1 | 1 | Transferred to SCI3nRDR | Framing error + parity error |
| 1* | 1 | 0 | 0 | Lost | Overrun error |
| 1* | 1 | 1 | 0 | Lost | Overrun error + framing error |
| 1* | 1 | 0 | 1 | Lost | Overrun error + parity error |
| 1* | 1 | 1 | 1 | Lost | Overrun error + framing error + parity error |

Note 1. In the case of an overrun error, the RDRF flag retains the state before the data reception.

Note: "+" indicates that two or more receive statuses occur simultaneously in a single reception operation.

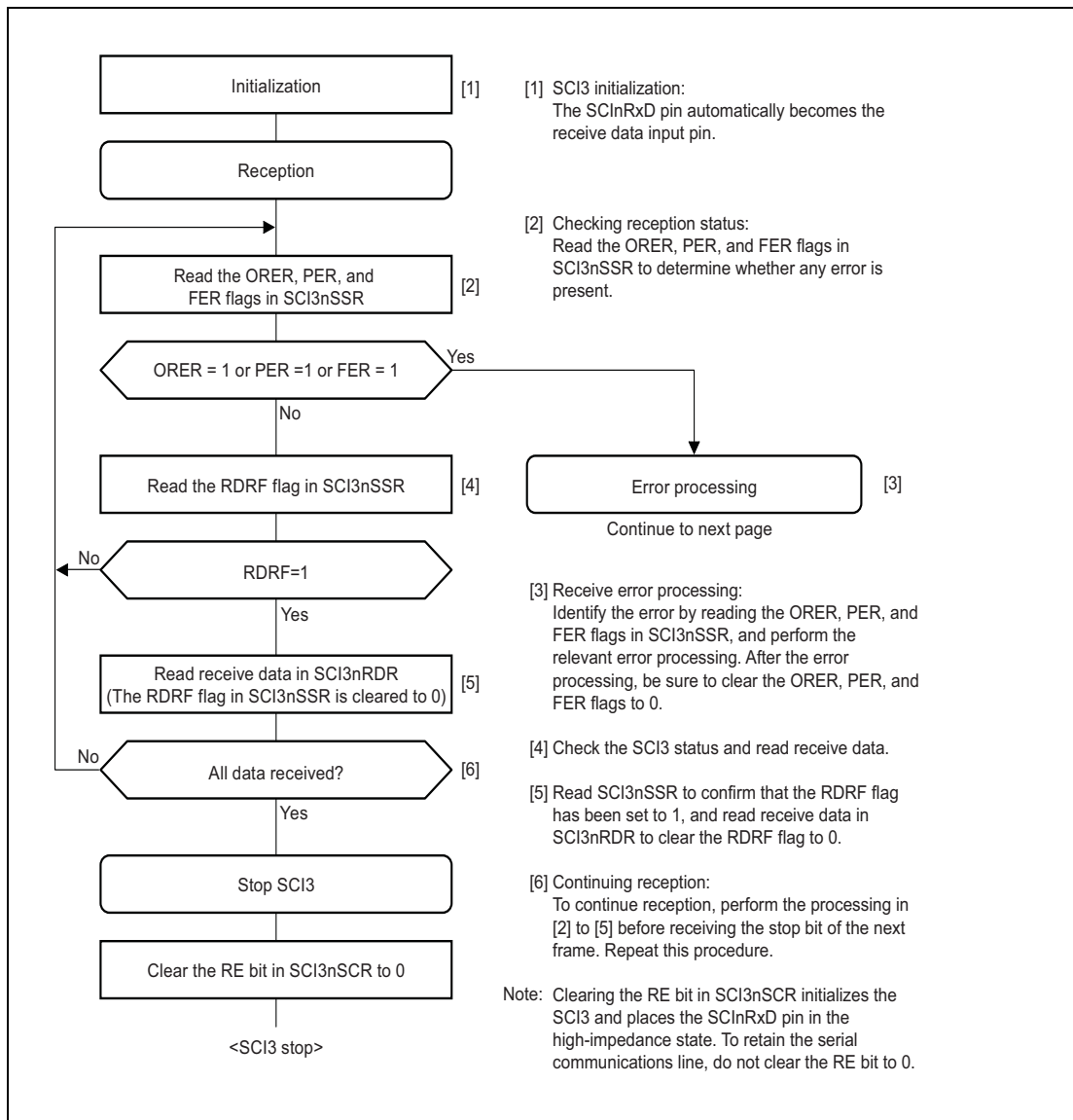


Figure 15.10 Example of Serial Reception Flowchart (1)

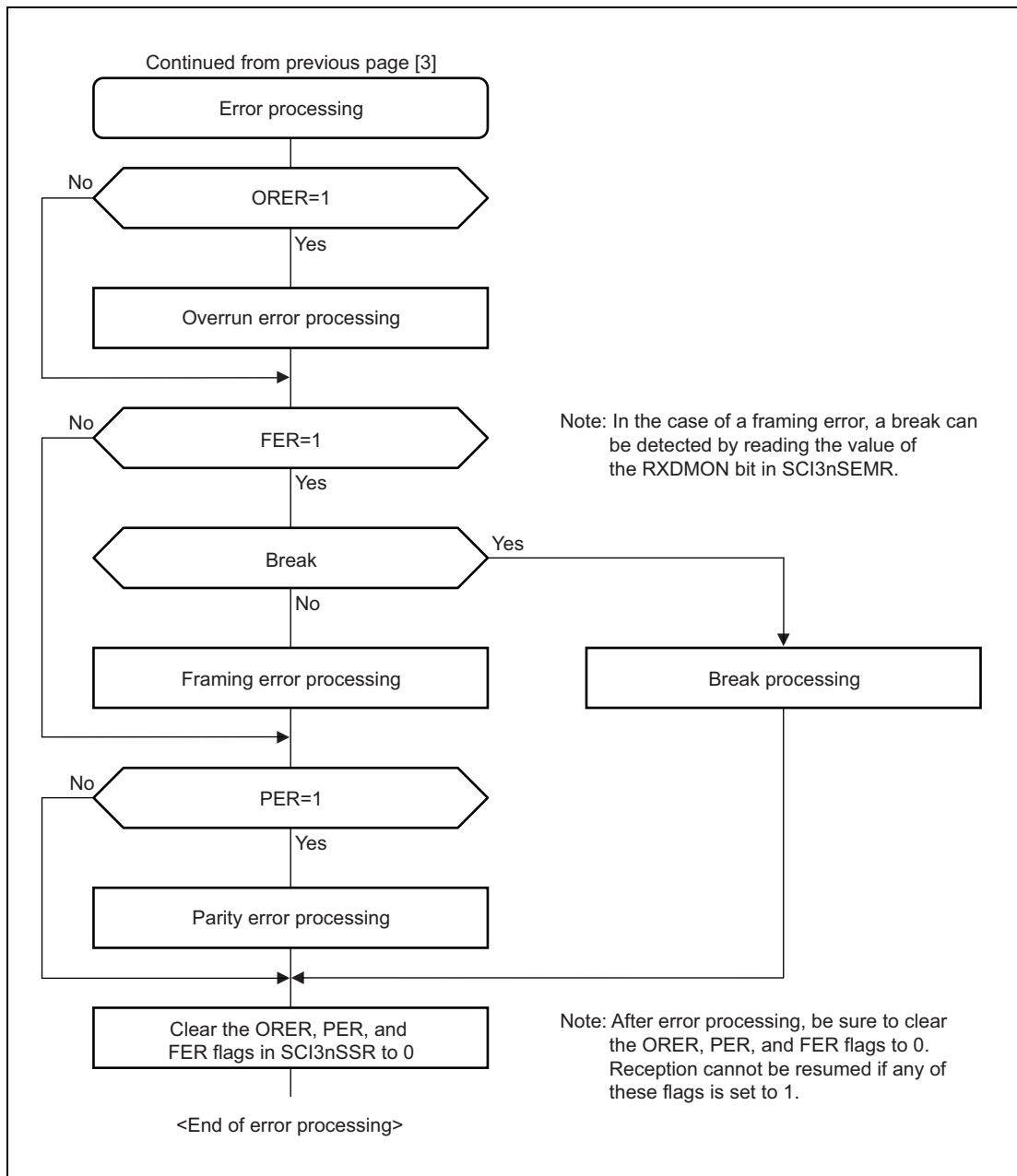


Figure 15.11 Example of Serial Reception Flowchart (2)

15.5 Multi-Processor Communication Function

15.5.1 Overview and Sample Connection

Using the multi-processor communication function allows data transmission and reception by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish the ID transmission cycle from the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle. When the multi-processor bit is set to 0, it indicates the data transmission cycle. **Figure 15.12** shows an example of communication between processors by using the multi-processor format. First, a transmitting station sends communication data in which the multi-processor bit (= 1) is added to the ID code of the receiving station. Next, the transmitting station sends communication data in which the multi-processor bit (= 0) is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself. When these IDs match, the receiving station receives communication data that is subsequently transmitted. If these IDs do not match, the receiving station skips communication data until it receives communication data in which the multi-processor bit is set to 1.

To support this function, the SCI3 provides the MPIE bit in SCI3nSCR. When the MPIE bit is set to 1, transfer of receive data from SCI3nRSR to SCI3nRDR, detection of a receive error, and setting the RDRF, FER, and ORER flags in SCI3nSSR are disabled until data in which the multi-processor bit is set to 1 is received. Upon receiving a character in which the multi-processor bit is set to 1, the MPB bit in SCI3nSSR is set to 1 and the MPIE bit is automatically cleared to 0, thus returning to a normal reception operation. When the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nRXI interrupt request is generated. While the MPIE bit is cleared to 0, reception operation is conducted regardless of the multi-processor bit value. The multi-processor bit is stored in the MPB bit in SCI3nSSR.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock that is used for multi-processor communications is also the same as the clock used in the normal asynchronous mode.

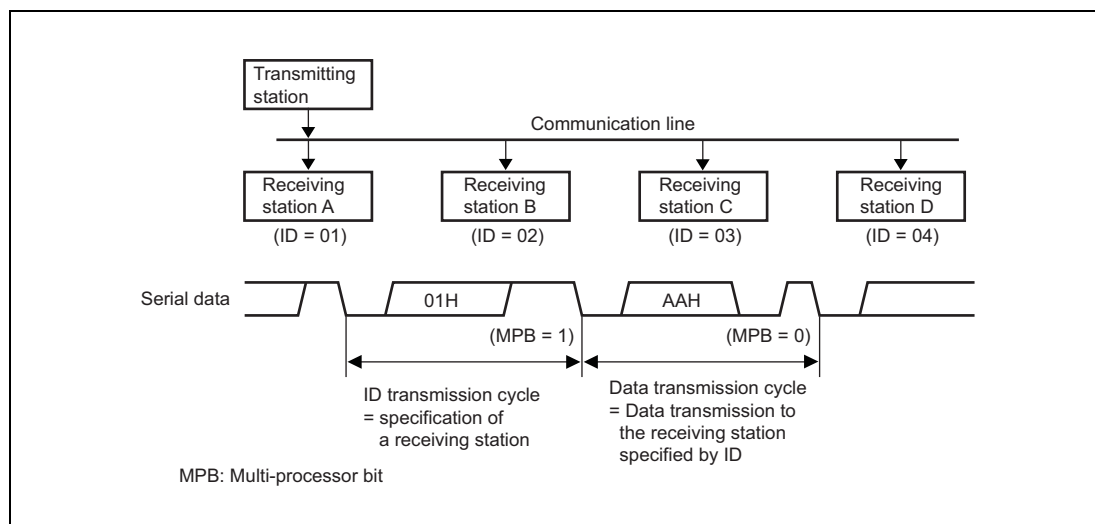


Figure 15.12 Example of Communication Using the Multi-Processor Format (Example of Transmission of Data AA_H to Receiving Station A)

15.5.2 Multi-Processor Serial Data Transmission

Figure 15.13 shows a sample flowchart of multi-processor data processing. In the ID transmission cycle, send the ID with the MPBT bit in SCI3nSSR set to 1. In the data transmission cycle, send data with the MPBT bit in SCI3nSSR cleared to 0. Other operations are the same as operations in asynchronous mode.

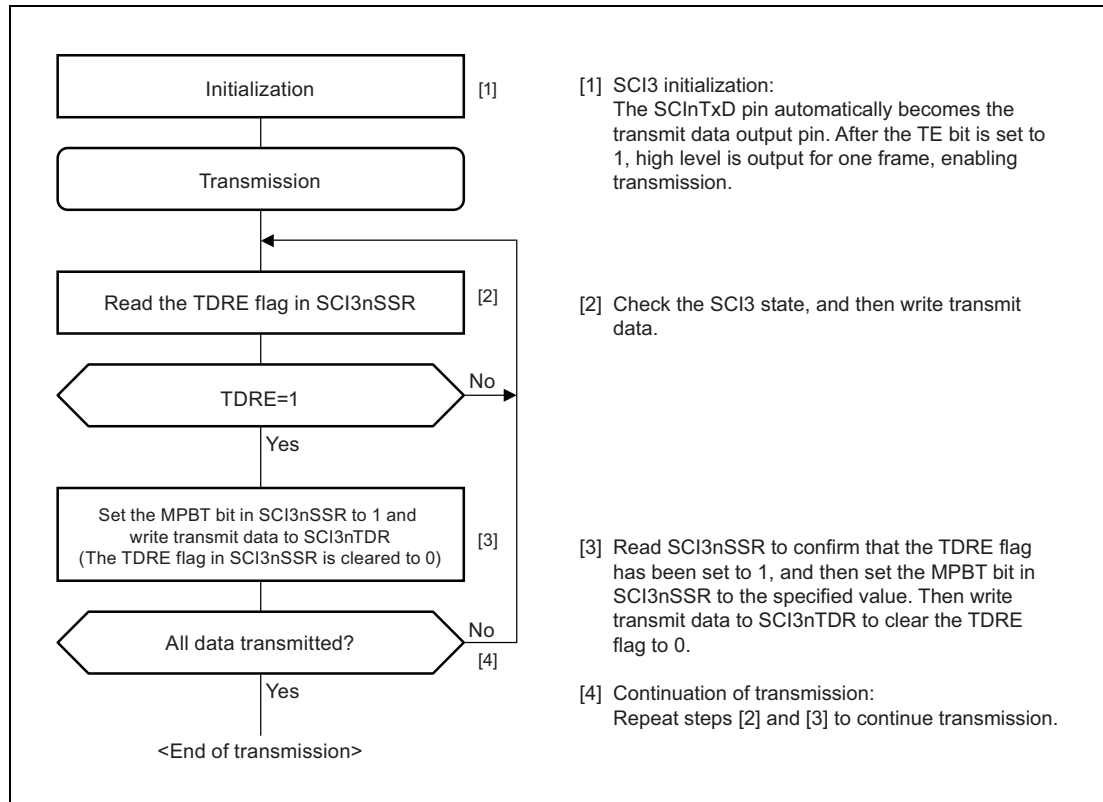


Figure 15.13 Example of Multi-Processor Serial Transmission Flowchart

15.5.3 Multi-Processor Serial Data Reception

Figure 15.15 shows a sample flowchart of multi-processor data reception. When the MPIE bit in SCI3nSCR is set to 1, reading the communication data is skipped until communication data in which the multi-processor bit is set to 1 is received. When communication data in which the multi-processor bit is set to 1 is received, the receive data is transferred to SCI3nRDR. At this time, an INTSCI3nRXI interrupt request is generated. Other operations are the same as operations in asynchronous mode.

Figure 15.14 shows an example of operation for reception.

CAUTION

Do not write data to SCI3nSCR when communication data in which the multi-processor bit is set to 1 is received. The MPIE bit may not become the desired state.

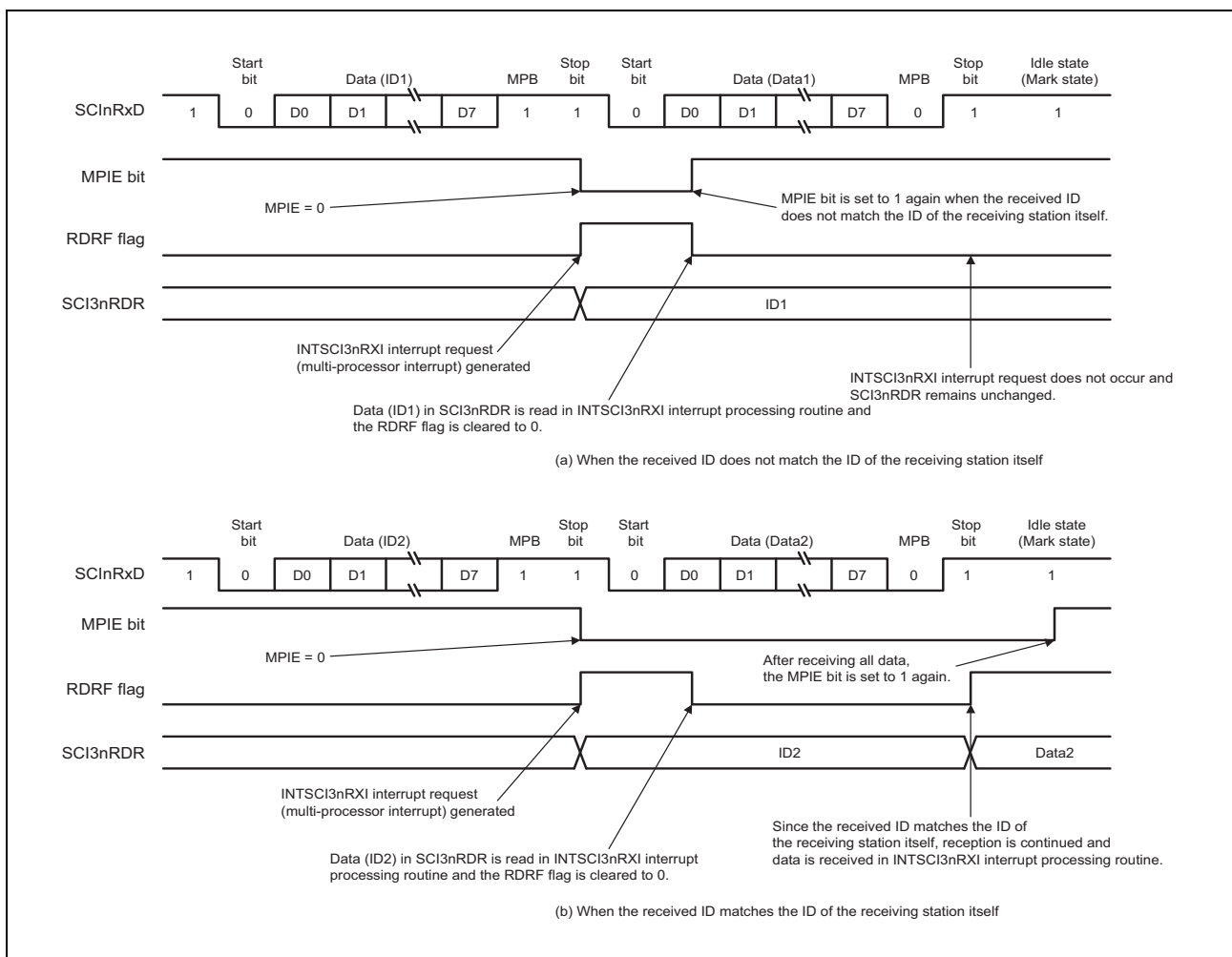


Figure 15.14 Example of SCI3 Reception (8-Bit Data, Multi-Processor Bit, One Stop Bit)

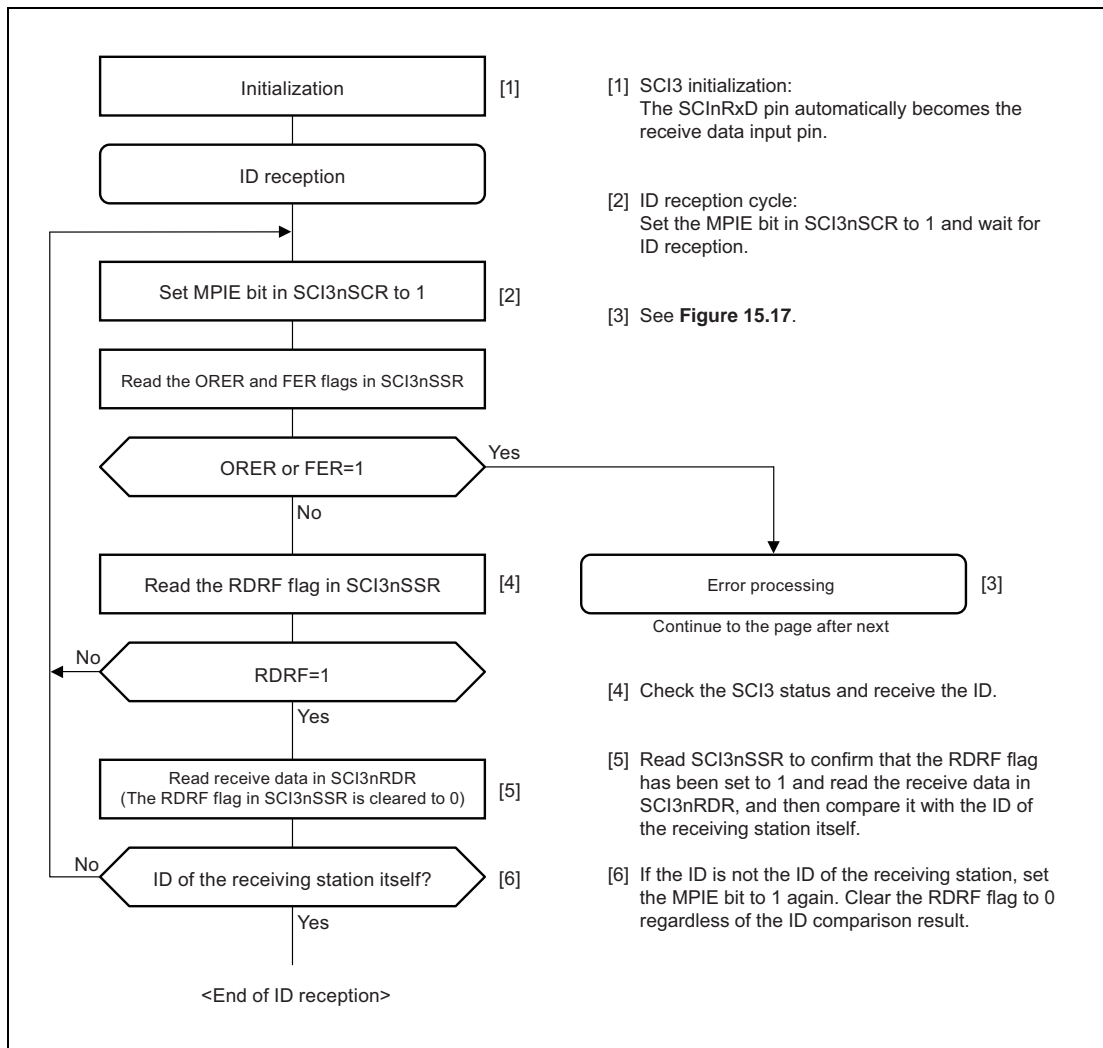


Figure 15.15 Example of Multi-Processor Serial Reception Flowchart (1)

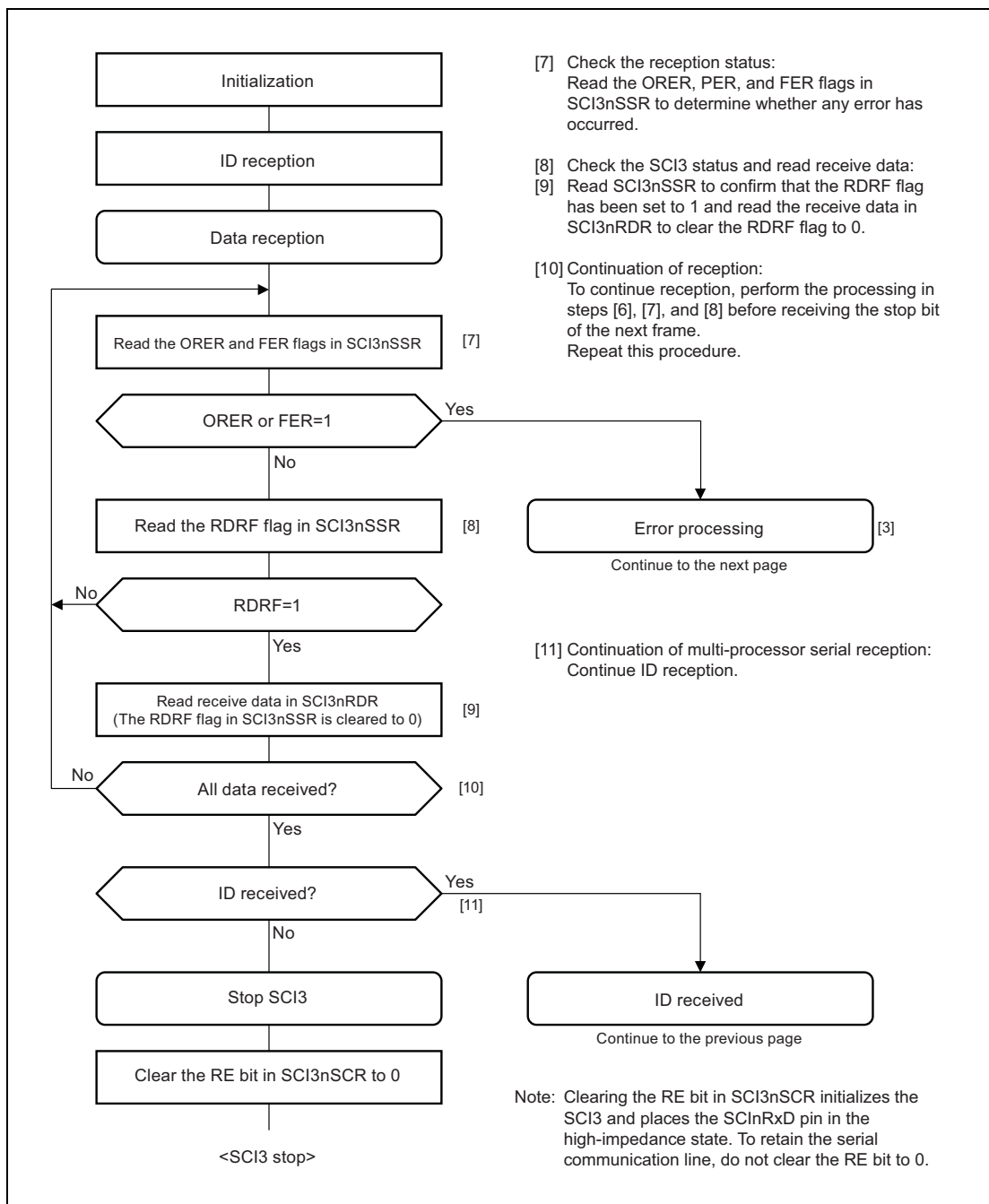


Figure 15.16 Example of Multi-Processor Serial Reception Flowchart (2)

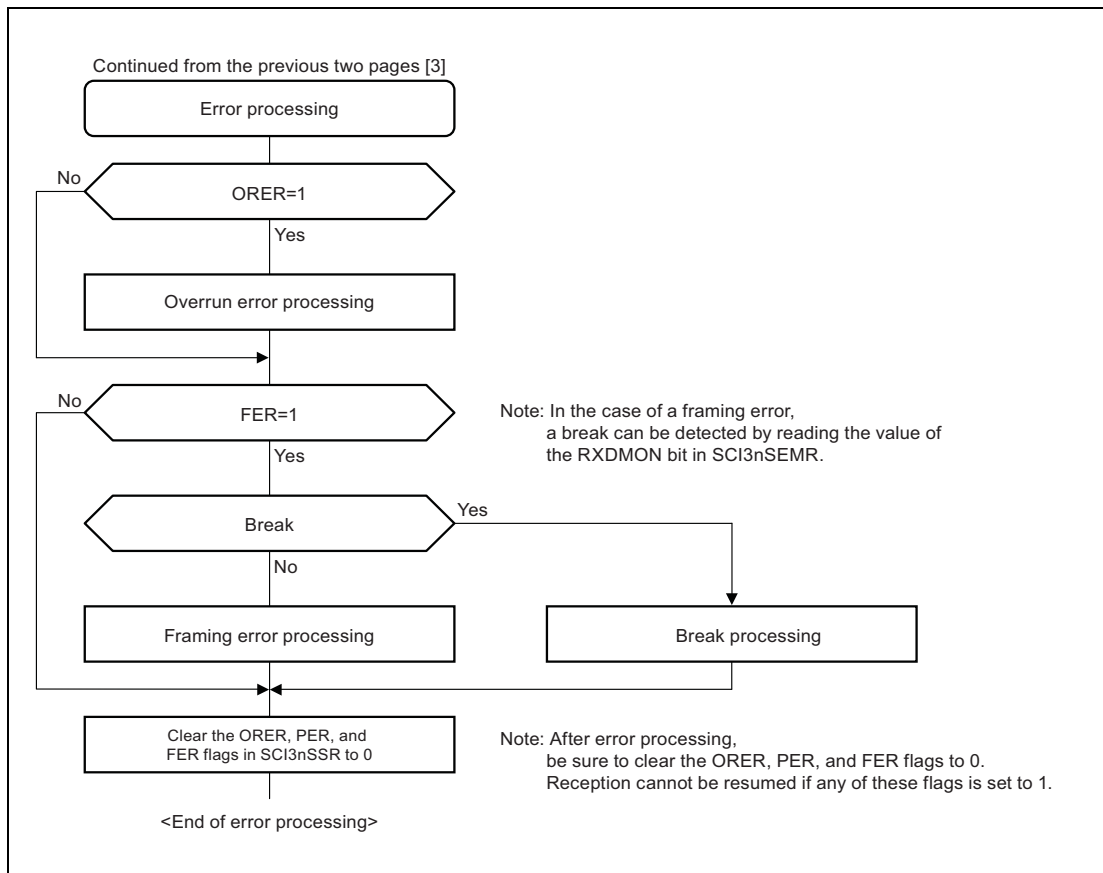


Figure 15.17 Example of Multi-Processor Serial Reception Flowchart (3)

15.6 Operation in Clock Synchronous Mode

Figure 15.18 shows the data format for clock synchronous serial data communication. In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In data transmission with the synchronization clock output, the SCI3 outputs data from one falling edge to the next falling edge of the synchronization clock. In data transmission with the synchronization clock input, the SCI3 outputs the first data (bit 0) after starting transfer immediately after clearing the TDRE bit in SCI3nSSR to 0, and then outputs the next bit data after 2 to 3 PCLK clock cycles from the rising edge of the synchronization clock. In data reception, the SCI3 receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last-bit output state. In clock synchronous mode, neither parity bit nor multi-processor bit can be added. The transmitter and the receiver are independent in the SCI3, enabling full-duplex communication by using a common clock. Both the transmitter and the receiver have a double-buffered structure so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

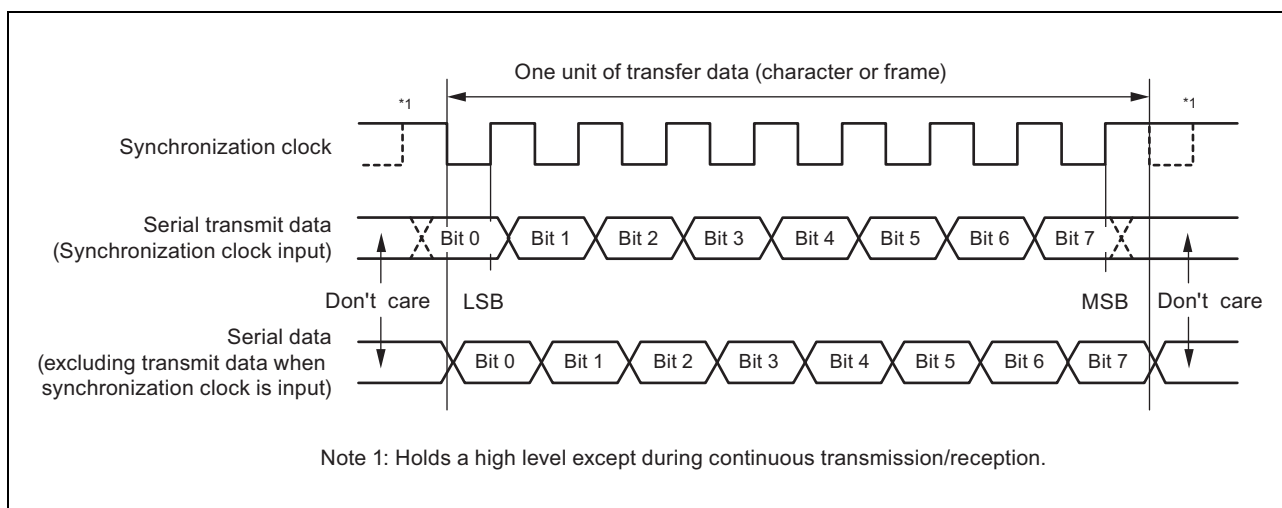


Figure 15.18 Data Format in Clock Synchronous Mode (LSB-First)

15.6.1 Clock

An internal clock generated by the on-chip baud rate generator or an external synchronous clock that is input from the SCInSCK pin can be selected by the setting of the CKE1 and CKE0 bits in SCI3nSCR. When operating the SCI3 on the internal clock, a synchronous clock is output from the SCInSCK pin. Eight pulses of the synchronous clock are output during transfer of one character, and the clock is held high while no data is transferred.

15.6.2 SCI3 Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, clear the TE and RE bits in SCI3nSCR to 0 and then initialize the SCI3 according to the sample flowchart in **Figure 15.19**. To switch the operation between transmission, reception, and transmission/reception, clear the TE and RE bits to 0 and then set these bits to the desired value. Before changing the transfer format, be sure to clear the TE and RE bits to 0. Note that clearing the TE bit to 0 sets the TDRE flag to 1, but clearing the RE bit to 0 initializes neither the RDRF, PER, FER, and ORER flags nor SCI3nRDR.

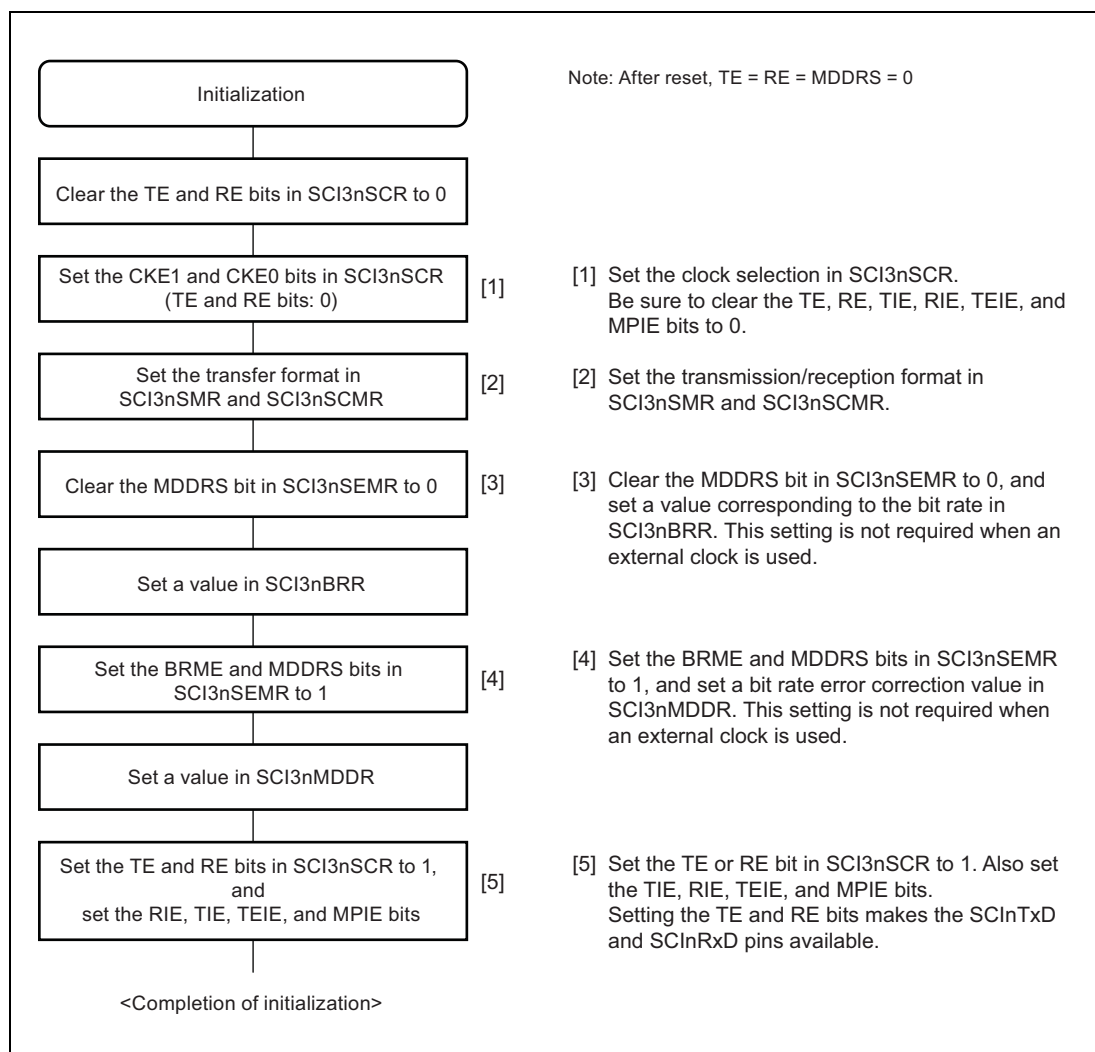


Figure 15.19 Example of SCI3 Initialization Flowchart

15.6.3 Serial Data Transmission (Clock Synchronous Mode)

Figure 15.20 shows an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the SCI3 operates as described below.

1. When transmit data is written to SCI3nTDR, the TDRE flag is automatically cleared to 0. The SCI3 monitors the TDRE flag in SCI3nSSR. When the flag is cleared, the SCI3 recognizes that data has been written to SCI3nTDR and transfers data from SCI3nTDR to SCI3nTSR, starting output of the first bit when the synchronization clock is input. To write transmit data to SCI3nTDR at a trigger of INTSCI3nTXI interrupt request, set the TIE bit to 1 and then set the TE bit to 1 or set both TIE and TE bits simultaneously with one instruction to generate a INTSCI3nTXI interrupt request for starting data transfer.
2. Transmission starts after data is transferred from SCI3nTDR to SCI3nTSR, and the TDRE flag is set to 1. When the TIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to SCI3nTDR in this INTSCI3nTXI interrupt processing routine before the transmission of the previously transferred data is completed. When a INTSCI3nTEI interrupt request is used, the TIE bit is cleared to 0 after the last transmit data has been written to SCI3nTDR, and the TEIE bit is set to 1.
3. 8-bit data is output from the SCInTxD pin in synchronization with the output clock (when clock output mode has been specified) or in synchronization with the input clock (when external clock has been specified).
4. The SCI3 checks the TDRE flag when the last bit is output.
5. When the TDRE flag is 0, the next transmit data is transferred from SCI3nTDR to SCI3nTSR, and serial transmission of the next frame starts.
6. When the TDRE flag is 1, the TEND flag in SCI3nSSR is set to 1 and the SCInSCK pin retains the output state of the last bit. If the TEIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTEI interrupt request is generated. The SCInSCK pin is held high.

Figure 15.21 shows a sample flowchart of serial data transmission. Also, **Figure 15.22** shows a sample flowchart for stopping the SCI3 after data transmission. Transmission will not start even if the TDRE flag is cleared while a receive error flag (ORER) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

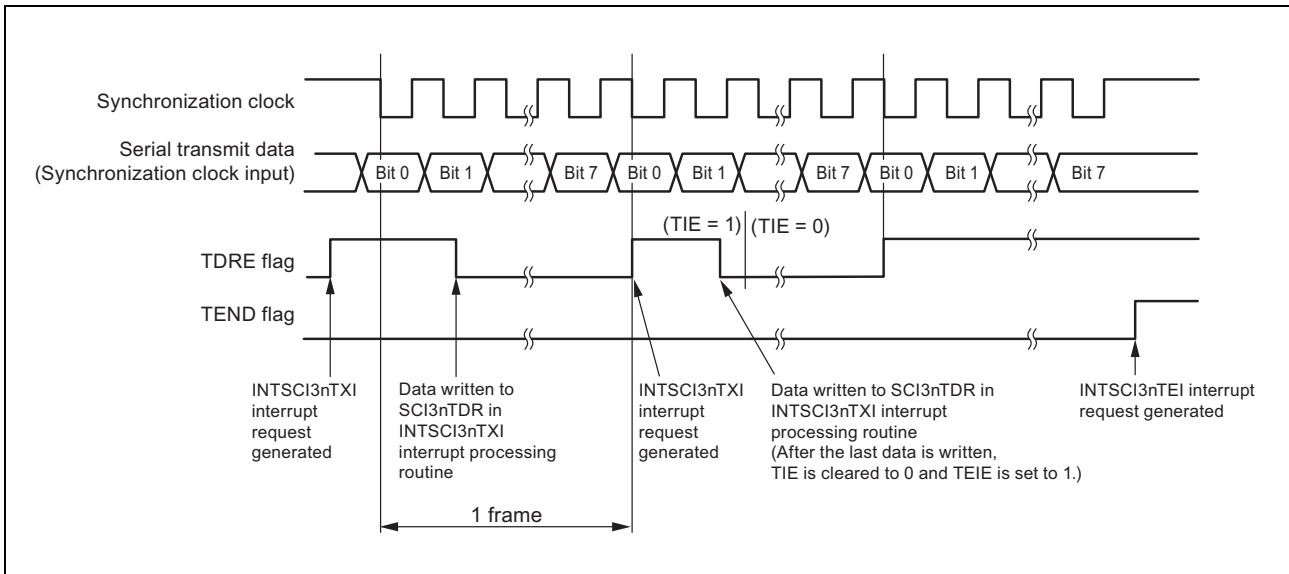


Figure 15.20 Example of Operation for Transmission in Clock Synchronous Mode

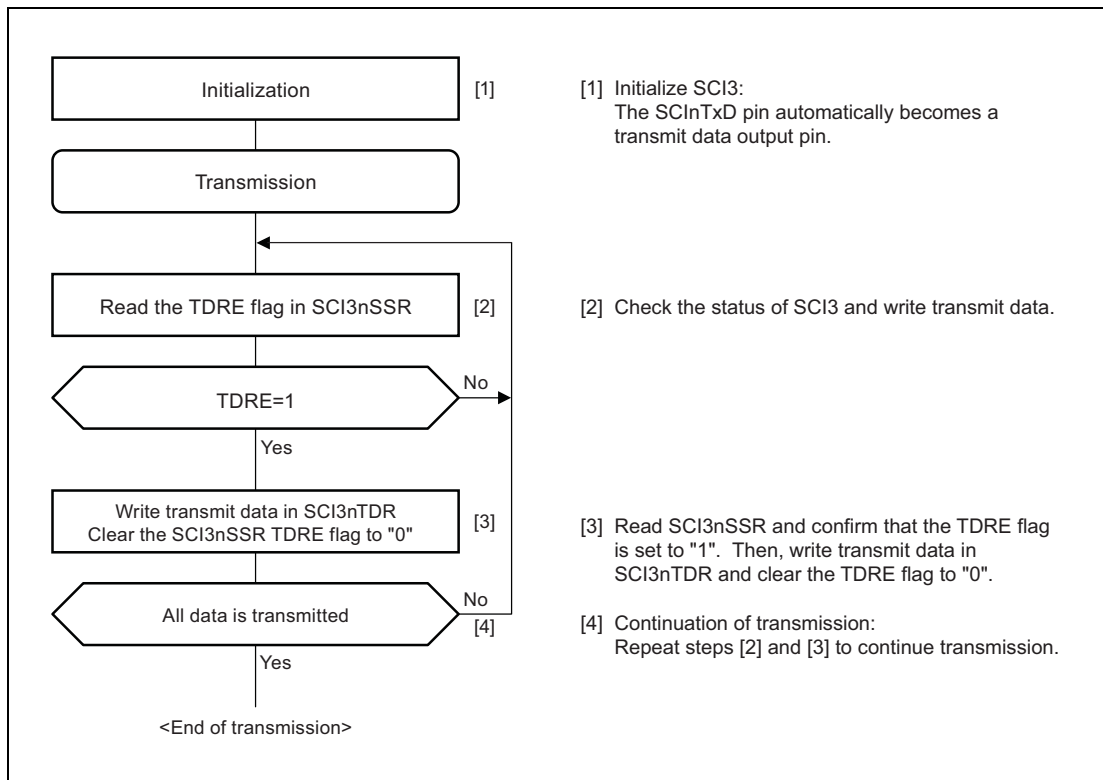


Figure 15.21 Example of Serial Transmission Flowchart

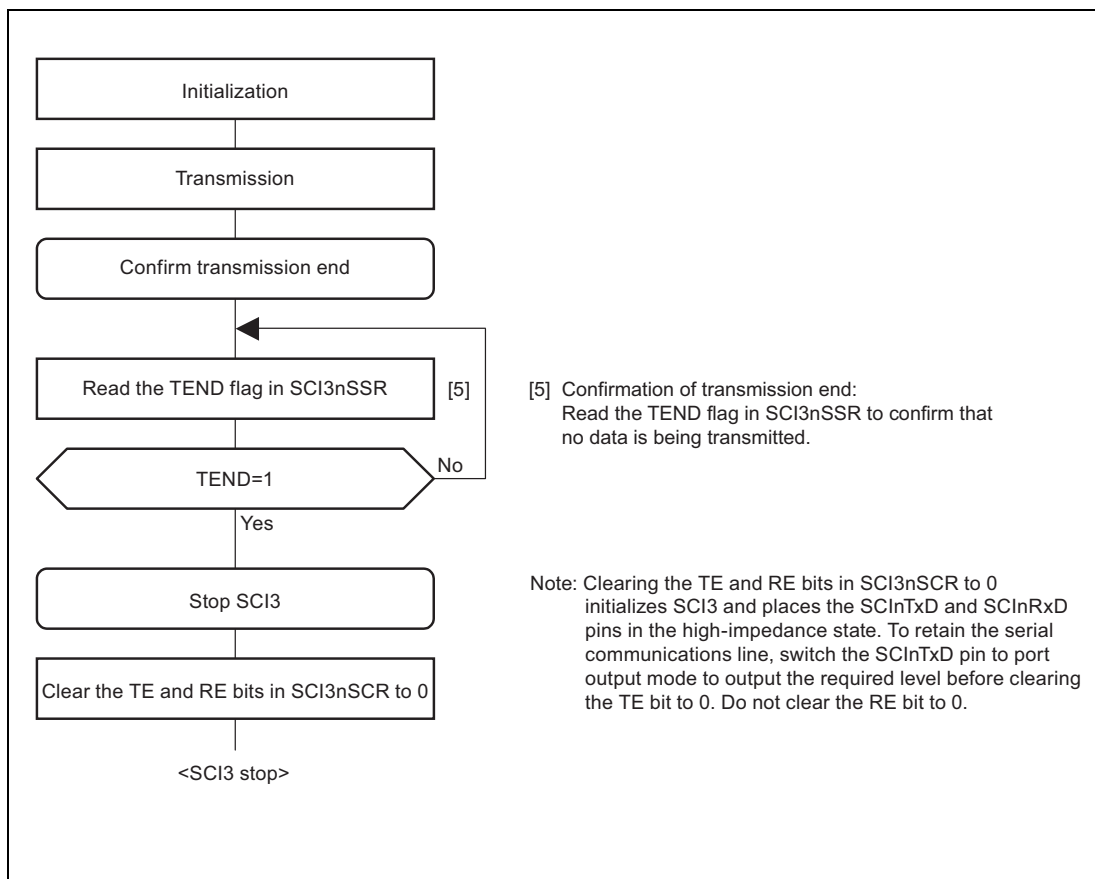


Figure 15.22 Example Flowchart for Stopping the SCI3 after Serial Transmission

15.6.4 Serial Data Reception (Clock Synchronous Mode)

Figure 15.23 shows an example of SCI3 operation for serial reception in clock synchronous mode. In serial data reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores receive data in SCI3nRSR.
2. When an overrun error occurs (the reception of the next data is completed with the RDRF flag in SCI3nSSR set to 1), the ORER flag in SCI3nSSR is set to 1. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated. Receive data is not transferred to SCI3nRDR. The RDRF flag retains the state of being set to 1.
3. When data has been successfully received, the RDRF flag in SCI3nSSR is set to 1 and the receive data is transferred to SCI3nRDR. When the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nRXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to SCI3nRDR in this INTSCI3nRXI interrupt processing routine before reception of the next data is completed. Reading SCI3nRDR automatically clears the RDRF flag to 0.

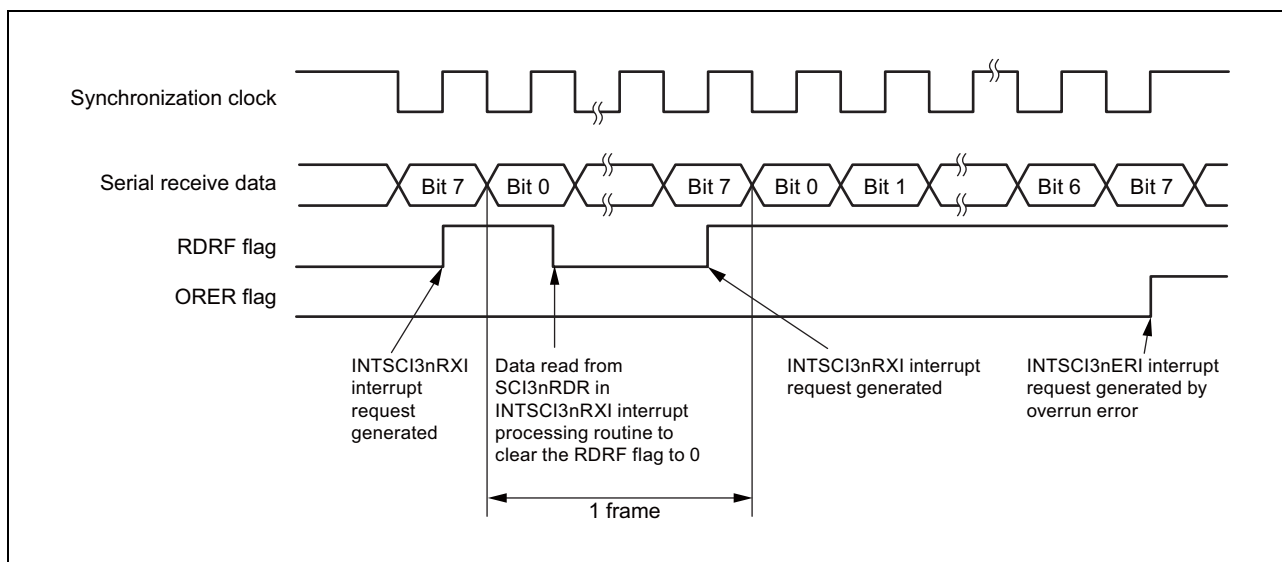


Figure 15.23 Example of SCI3 Operation for Reception

Subsequent transmission and reception are disabled with a receive error flag set to 1. Therefore, be sure to clear the ORER, FER, PER, and RDRF flags to 0 before continuing reception. **Figure 15.24** shows an example of flowchart for data reception.

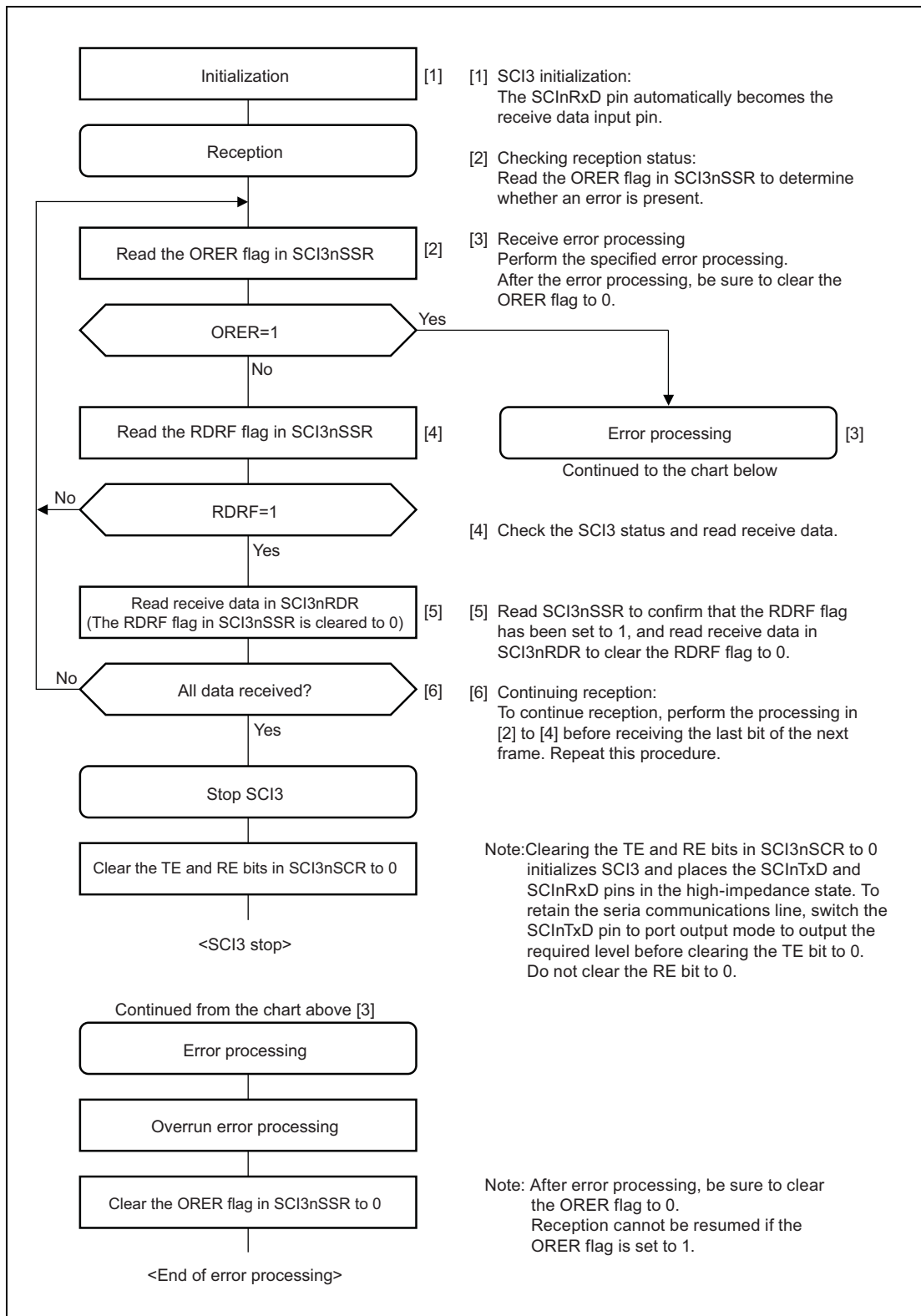


Figure 15.24 Example of Serial Reception Flowchart

15.6.5 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 15.25 shows a sample flowchart for simultaneous data transmit and receive operations. After the SCI3 is initialized, perform the following procedure for simultaneous data transmit and receive operations.

1. To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1. Then clear the TE bit to 0 and then set the TE and RE bits to 1 with a single instruction.
2. To switch from receive mode to simultaneous transmit and receive mode, check that the SCI3 has finished reception and clear the RE bit to 0. Then check that the RDRF and error flags (ORER, FER, and PER) are cleared to 0 and then set the TE and RE bits to 1 with a single instruction.

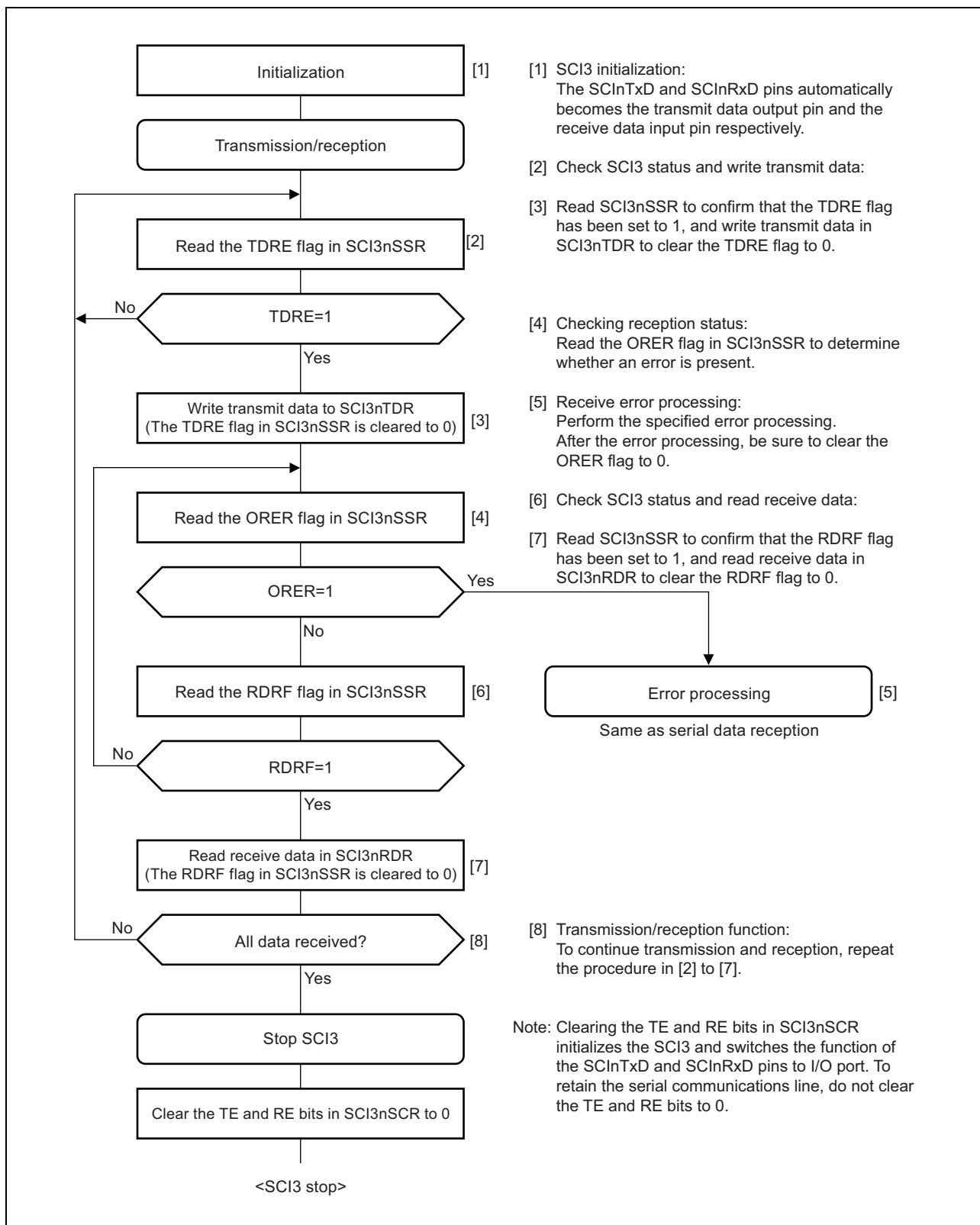


Figure 15.25 Example of Simultaneous Serial Transmission and Reception Flowchart

15.7 Bit Rate Modulation Function

The bit rate modulation function corrects a bit rate by averagely enabling the internal clocks specified by the CKS1 and CKS0 bits in SCI3nSMR for the number specified by SCI3nMDDR out of 256 clocks.

Figure 15.26 shows an example of asynchronous mode in which PCLK clock is selected with the CKS1 and CKS0 bits, SCI3nBRR is set to 0, and SCI3nMDDR is set to 160. In this example, the reference clock cycle is corrected to 256/160 on average and the bit rate is corrected to 160/256. Note that the pulse widths of the internal reference clock expand or contract for the amount of the selected internal clocks because there is a deviation in the enabling of the internal clocks.

Do not use this function with the maximum speed settings (CKS1 and CKS0 bits = 0 in SCI3nSMR, CKE1 bit = 0 in SCI3nSCR, and SCI3nBRR = 0) for clock synchronous mode.

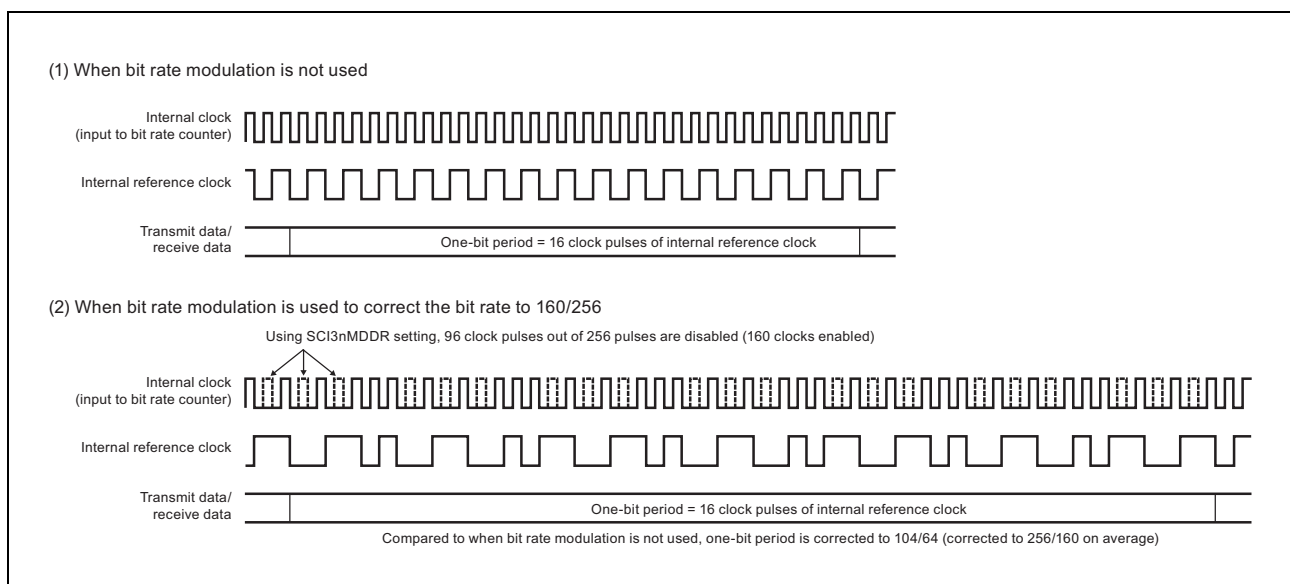


Figure 15.26 Example of Internal Reference Clock when the Bit Rate Modulation Function is Used

15.8 Interrupt Sources

Table 15.25 lists interrupt sources. Each interrupt source outputs an individual interrupt request signal. These interrupt sources can be enabled independently with the enable bit in SCI3nSCR.

A INTSCI3nTXI interrupt request is generated when the TDRE flag in SCI3nSSR is set to 1. A INTSCI3nTEI interrupt request is generated when the TEND flag in SCI3nSSR is set to 1. A INTSCI3nTXI interrupt request can activate the DMAC to handle data transfer. The TDRE flag is automatically cleared to 0 when data is transferred using the DMAC.

CAUTION

The TDRE flag and TEND flag cannot be cleared to 0 when the TE bit in SCI3nSCR is 0. The TEND flag is the level interrupt request flag for the INTSCI3nTEI interrupt. Therefore, when the TE bit is 0, do not set the TEIE bit in SCI3nSCR to 1.

An INTSCI3nRXI interrupt request is generated when the RDRF flag in SCI3nSSR is set to 1. An INTSCI3nERI interrupt is generated when any of the ORER, PER, and FER flags in SCI3nSSR is set to 1. An INTSCI3nRXI interrupt request can activate the DMAC to handle data transfer. The RDRF flag is automatically cleared to 0 when data is transferred using the DMAC.

A INTSCI3nTEI interrupt request is generated when the TEND flag is set to 1 with the TEIE bit set to 1.

CAUTION

If a INTSCI3nTEI interrupt request and a INTSCI3nTXI interrupt request are generated at the same time, the INTSCI3nTXI interrupt request is accepted first. Note that clearing the TDRE flag to 0 at this time in the INTSCI3nTXI interrupt processing routine also clears the TEND flag to 0 automatically, disabling the branch to the INTSCI3nTEI interrupt processing routine.

Table 15.25 SCI3 Interrupt Sources

| Name | Interrupt Source | Interrupt Flag | DMAC Activation | DTS Activation |
|-------------|---------------------|----------------|-----------------|----------------|
| INTSCI3nERI | Receive error | ORER, FER, PER | Not possible | Not possible |
| INTSCI3nRXI | Receive data full | RDRF | Possible | Possible |
| INTSCI3nTXI | Transmit data empty | TDRE | Possible | Possible |
| INTSCI3nTEI | Transmit end | TEND | Not possible | Not possible |

15.9 Usage Notes

15.9.1 Break Detection and Processing

A break can be detected by reading the RXDMON bit in SCI3nSEMR when detecting a framing error. Since all inputs from the SCInRxD pin are 0 in the break state, the FER flag is set to 1 and the PER flag may also be set to 1. The SCI3 continues data reception even after it receives a break. For this reason, note that the FER flag is set to 1 again even after the FER flag is cleared to 0.

15.9.2 Mark State and Break Output

While the TE bit is 0 (transmission/reception disabled), the SCInTxD pin can output any level by switching the SCInTxD pin to a general output port. This allows the SCInTxD pin to enter the mark state or to output a break during data transmission. When the TE bit is cleared to 0, the transmitter unit is initialized regardless of the current transmission state.

15.9.3 Receive Error Flags and Transmit Operations in Clock Synchronous Mode

During the clock synchronous simultaneous data transmit/receive operation, transmission cannot be started when a receive error flag (ORER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note that the receive error flags cannot be cleared to 0 even by clearing the RE bit to 0.

15.9.4 Relationship between Writing to SCI3nTDR and the TDRE Flag

The TDRE flag in SCI3nSSR is a status flag indicating that transmit data in SCI3nTDR has been transferred to SCI3nTSR. The TDRE flag is set to 1 when the SCI3 transfers data from SCI3nTDR to SCI3nTSR.

Data can be written to SCI3nTDR regardless of the status of the TDRE flag. However, writing new data to SCI3nTDR while the TDRE flag is 0 will cause the data stored in SCI3nTDR to be lost because it has not been transferred to SCI3nTSR. Therefore, make sure to check that the TDRE flag is set to 1 before writing transmit data to SCI3nTDR.

15.9.5 Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode

When using an external synchronization clock, clear the TDRE flag to 0 and then input a transmission clock (See **Figure 15.27**). Also in continuous transmission mode, clear the TDRE flag to 0 and then input a transmission clock for the next frame.

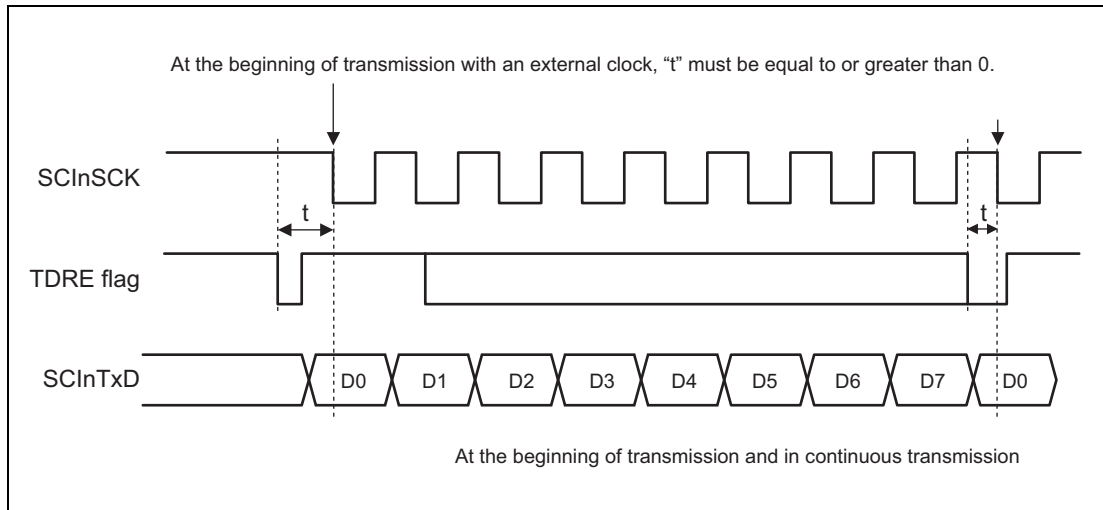


Figure 15.27 Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode

15.9.6 External Clock Input in Clock Synchronous Mode

For the external clock SCInSCK input in clock synchronous mode, see **Section 37.9.7, SCI3 Timing**.

Section 16 LIN/UART Interface (RLIN3)

This section contains a generic description of the LIN/UART interface (RLIN3).

The first part of this section describes all RH850/P1M-E specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN3.

16.1 Features of RH850/P1M-E RLIN3

16.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN3 units.

Each RLIN3 unit has a single channel interface. "Number of channels" therefore has the same meaning as "number of units" in this section.

Table 16.1 Number of Units

| Product Name | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of units | 2 | 2 |
| Name | RLIN3n (n = 0, 1) | RLIN3n (n = 0, 1) |

Table 16.2 Unit Configurations and Channels

| Unit Name (Channel Name) | Channels per Unit | RH850/P1M-E 100 pins (2 ch) | RH850/P1M-E 144 pins (2 ch) |
|-----------------------------|----------------------|-----------------------------------|-----------------------------------|
| RLIN30 | 1 | √ | √ |
| RLIN31 | 1 | √ | √ |

Note: The channel names are same as those of the corresponding units.

Table 16.3 Index

| Index | Description |
|-------|---|
| n | Throughout this section, the individual RLIN3 units are identified by the index "n" (n = 0): for example, RLIN3nLCUC is the LIN control register. |
| b | Throughout this section, the individual transmit/receive data buffers of RLIN3n are identified by the index "b" (b = 1 to 8): for example, RLIN3nLDBR1 is the first stage data buffer register. |

The following lists the index corresponding to each product.

Table 16.4 Index Correspondence of Each Product

| Index Corresponding to Product |
|--------------------------------|
| All products |
| b = 1 to 8 |

16.1.2 Register Base Address

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses in general.

Table 16.5 Register Base Addresses

| Base Address Name | Base Address |
|-------------------|------------------------|
| <RLIN30_base> | FFDF 8000 _H |
| <RLIN31_base> | FFDF 9000 _H |

16.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

Table 16.6 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|---------------------------------|-------------------------------------|
| RLIN3n | LIN communication clock sources | High-speed peripheral clock CLK_HSB |

16.1.4 Interrupt Request

RLIN3 interrupt requests are listed in the following table.

Table 16.7 Interrupt Requests

| Unit Interrupt Signal | Description | Interrupt Number | DMA Trigger Number |
|-----------------------|-------------------------------------|------------------|--------------------|
| RLIN30 | | | |
| INTRLIN3nUR2 (n = 0) | RLIN30 status interrupt | 114 | — |
| INTRLIN3nUR1 (n = 0) | RLIN30 receive completion interrupt | 115 | 95 |
| INTRLIN3nUR0 (n = 0) | RLIN30 transmit interrupt | 116 | 96 |
| RLIN31 | | | |
| INTRLIN3nUR2 (n = 1) | RLIN31 status interrupt | 117 | — |
| INTRLIN3nUR1 (n = 1) | RLIN31 receive completion interrupt | 118 | 97 |
| INTRLIN3nUR0 (n = 1) | RLIN31 transmit interrupt | 119 | 98 |

16.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

16.1.6 External Input/output Signals

External input/output signals of RLIN3 are listed below.

Table 16.8 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|------------------|-----|-----------------------------|----------------------------------|
| RLIN30 | | | |
| RLIN3nRX (n = 0) | I | RLIN30 receive data input | RLIN30RX |
| RLIN3nTX (n = 0) | O | RLIN30 transmit data output | RLIN30TX |
| RLIN31 | | | |
| RLIN3nRX (n = 1) | I | RLIN31 receive data input | RLIN31RX |
| RLIN3nTX (n = 1) | O | RLIN31 transmit data output | RLIN31TX |

16.1.7 Combination of Pin Name and Port Name

Combinations of RLIN3 pins and ports are listed in the following table.

Table 16.9 Combination of Pins and Ports

| Function | Pin Name | Port Name | | | | | |
|----------|----------|--------------------|---------|---------------------|---------------------|---------------------|---------|
| | | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 |
| RLIN30 | RLIN30RX | P1_1 | P3_4 | P4_0 | P5_10 | P5_14 | P5_14 |
| | RLIN30TX | P1_0 ^{*1} | P3_5 | P4_1 | P5_11 ^{*1} | P5_15 ^{*1} | P0_0 |
| RLIN31 | RLIN31RX | P2_5 | P0_1 | P5_15 ^{*1} | P5_12 ^{*1} | — | — |
| | RLIN31TX | P2_4 | P3_11 | P3_11 | P5_13 ^{*1} | — | — |

Note 1. Usable in the 144-pin product

16.2 Overview

16.2.1 Functional Overview

The LIN/UART interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602, and automatically performs frame communication and error determination.

The LIN/UART interface is provided with UART mode and can also be used as a UART.

The appropriate mode should be used for the LIN/UART interface according to the application: LIN master, or UART.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
 - LIN wake-up mode
 - LIN operation mode
- LIN self-test mode

UART

- LIN reset mode
- UART mode

Table 16.10 gives the LIN/UART interface specifications.

Table 16.10 LIN/UART Interface Specifications (1/2)

| Item | Specifications | |
|----------------------------|------------------------------------|--|
| Number of Channel | 2 channels | |
| LIN communication function | Protocol | LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602 |
| | Variable frame structure | Master <ul style="list-style-type: none"> • Break transmission width: 13 to 28 Tbits • Break delimiter transmission width: 1 to 4 Tbits • Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)*¹ • Transmission response space width: 0 to 7 Tbits*¹ • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits |
| | Checksum | <ul style="list-style-type: none"> • Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame) |
| | Response field data byte count | Variable from 0 to 8 bytes Multi-byte (9 or more bytes) response transmission and reception also possible |
| | Frame communication mode | Master <ul style="list-style-type: none"> • Mode in which header transmission and response transmission/reception are started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode) |
| | Wake-up transmission and reception | LIN wake-up mode provided <ul style="list-style-type: none"> • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured |
| | Status | Master <ul style="list-style-type: none"> • Frame/wake-up transmission complete • Header transmission complete • Frame/wake-up reception complete*² • Data 1 reception complete • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode) |
| | Error status | Master <ul style="list-style-type: none"> • Bit error • Checksum error • Frame timeout error/response timeout error • Physical bus error • Framing error • Response preparation error |
| | Baud rate selection | Baud rate conforming to the LIN specifications generated using baud rate generator |
| | Test mode | Self-test mode for user evaluation |
| | Interrupt function | Master <ul style="list-style-type: none"> • Header/frame/wake-up transmission complete • Frame/wake-up reception complete*² • Error detection |

Table 16.10 LIN/UART Interface Specifications (2/2)

| Item | Specifications |
|--|---|
| UART communication function | Data buffer <ul style="list-style-type: none"> • Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1. Character length of 7, 8, and 9 bits supported) • UART buffer (exclusively for transmission; variable data length from 1 to 9 bits. Character length of 7 and 8 bits supported) • Reception data buffer (exclusively for reception; data length of 1. Character length of 7, 8, and 9 bits supported) |
| | Data format <hr/> Character length: 7 or 8 bits Length of 9 bits supported by using the expansion bit. <hr/> Transmission stop bit: 1 or 2 bits <hr/> Parity function: odd, even, 0, or none <hr/> LSB- or MSB-first transfer selectable <hr/> Reverse input/output of transmission/reception data possible |
| | Status <ul style="list-style-type: none"> • Transmission status • Reception status • Successful UART buffer transmission • Error detection • Expansion bit detection • ID match • Reset mode status |
| | Error status <ul style="list-style-type: none"> • Bit error • Framing error • Parity error • Overrun error |
| | Baud rate selection <hr/> With the built-in baud rate generator, any baud rate can be set. <hr/> When a certain expansion bit is at the expected level, the data received can be compared to the 8-bit data preset in the register. <hr/> The stop bit received is guaranteed (start of transmission can be delayed when start of transmission is attempted during reception of the stop bit). |
| Interrupt function <ul style="list-style-type: none"> • Transmission start/complete • Reception complete • Status/error detection | |

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the input signal low-level width count is indicated.

16.2.2 Block Diagram

Figure 16.1 shows a block diagram of the LIN/UART interface.

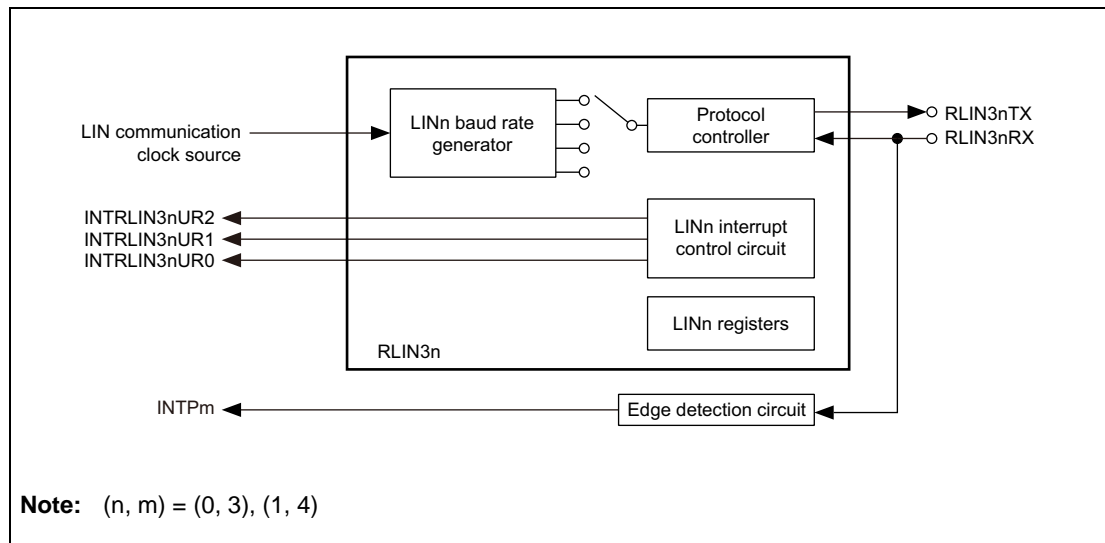


Figure 16.1 LIN/UART Interface Block Diagram

16.2.3 Description of Blocks

- RLIN3nTX, RLIN3nRX: LIN/UART interface I/O pins
- LINn baud rate generator: Generates the LIN/UART interface communication clock
- LINn registers: LIN/UART interface registers
- LINn interrupt control circuit: Controls interrupt requests generated by the LIN/UART interface

16.3 Registers

16.3.1 List of Registers

RLIN3 registers are listed in the following table.

For <RLIN3n_base>, see **Section 16.1.2, Register Base Address**.

Table 16.11 Registers

| Module | Register | Symbol | Address | LIN Master | UART |
|--------|---|--------------|---------------------------------|------------|------|
| RLIN3n | LIN wake-up baud rate selector register | RLN3nLWBR | <RLIN3n_base> + 01 _H | √ | √ |
| RLIN3n | LIN / UART baud rate prescaler 01 register | RLN3nLBRP01 | <RLIN3n_base> + 02 _H | — | √ |
| RLIN3n | LIN / UART baud rate prescaler 0 register | RLN3nLBRP0 | <RLIN3n_base> + 02 _H | √ | √ |
| RLIN3n | LIN / UART baud rate prescaler 1 register | RLN3nLBRP1 | <RLIN3n_base> + 03 _H | √ | √ |
| RLIN3n | LIN self-test control register | RLN3nLSTC | <RLIN3n_base> + 04 _H | √ | — |
| RLIN3n | LIN / UART mode register | RLN3nLMD | <RLIN3n_base> + 08 _H | √ | √ |
| RLIN3n | LIN break field configuration register/ UART configuration register | RLN3nLBFC | <RLIN3n_base> + 09 _H | √ | √ |
| RLIN3n | LIN / UART space configuration register | RLN3nLSC | <RLIN3n_base> + 0A _H | √ | √ |
| RLIN3n | LIN wake-up configuration register | RLN3nLWUP | <RLIN3n_base> + 0B _H | √ | — |
| RLIN3n | LIN interrupt enable register | RLN3nLIE | <RLIN3n_base> + 0C _H | √ | — |
| RLIN3n | LIN / UART error detection enable register | RLN3nLEDE | <RLIN3n_base> + 0D _H | √ | √ |
| RLIN3n | LIN / UART control register | RLN3nLCUC | <RLIN3n_base> + 0E _H | √ | √ |
| RLIN3n | LIN / UART transmission control register | RLN3nLTRC | <RLIN3n_base> + 10 _H | √ | √ |
| RLIN3n | LIN / UART mode status register | RLN3nLMST | <RLIN3n_base> + 11 _H | √ | √ |
| RLIN3n | LIN / UART status register | RLN3nLST | <RLIN3n_base> + 12 _H | √ | √ |
| RLIN3n | LIN / UART error status register | RLN3nLEST | <RLIN3n_base> + 13 _H | √ | √ |
| RLIN3n | LIN data field configuration register | RLN3nLDFC | <RLIN3n_base> + 14 _H | √ | √ |
| RLIN3n | LIN / UART ID buffer register | RLN3nLIDB | <RLIN3n_base> + 15 _H | √ | √ |
| RLIN3n | LIN checksum buffer register | RLN3nLCBR | <RLIN3n_base> + 16 _H | √ | — |
| RLIN3n | UART data buffer 0 register | RLN3nLUDB0 | <RLIN3n_base> + 17 _H | — | √ |
| RLIN3n | LIN / UART data buffer 1 register | RLN3nLDBR1 | <RLIN3n_base> + 18 _H | √ | √ |
| RLIN3n | LIN / UART data buffer 2 register | RLN3nLDBR2 | <RLIN3n_base> + 19 _H | √ | √ |
| RLIN3n | LIN / UART data buffer 3 register | RLN3nLDBR3 | <RLIN3n_base> + 1A _H | √ | √ |
| RLIN3n | LIN / UART data buffer 4 register | RLN3nLDBR4 | <RLIN3n_base> + 1B _H | √ | √ |
| RLIN3n | LIN / UART data buffer 5 register | RLN3nLDBR5 | <RLIN3n_base> + 1C _H | √ | √ |
| RLIN3n | LIN / UART data buffer 6 register | RLN3nLDBR6 | <RLIN3n_base> + 1D _H | √ | √ |
| RLIN3n | LIN / UART data buffer 7 register | RLN3nLDBR7 | <RLIN3n_base> + 1E _H | √ | √ |
| RLIN3n | LIN / UART data buffer 8 register | RLN3nLDBR8 | <RLIN3n_base> + 1F _H | √ | √ |
| RLIN3n | UART operation enable register | RLN3nLUOER | <RLIN3n_base> + 20 _H | — | √ |
| RLIN3n | UART option register 1 | RLN3nLUOR1 | <RLIN3n_base> + 21 _H | — | √ |
| RLIN3n | UART transmission data register | RLN3nLUTDR | <RLIN3n_base> + 24 _H | — | √ |
| RLIN3n | UART transmission data register L | RLN3nLUTDRL | <RLIN3n_base> + 24 _H | — | √ |
| RLIN3n | UART transmission data register H | RLN3nLUTDRH | <RLIN3n_base> + 25 _H | — | √ |
| RLIN3n | UART reception data register | RLN3nLURDR | <RLIN3n_base> + 26 _H | — | √ |
| RLIN3n | UART reception data register L | RLN3nLURDRL | <RLIN3n_base> + 26 _H | — | √ |
| RLIN3n | UART reception data register H | RLN3nLURDRH | <RLIN3n_base> + 27 _H | — | √ |
| RLIN3n | UART wait transmission data register | RLN3nLUWTDR | <RLIN3n_base> + 28 _H | — | √ |
| RLIN3n | UART wait transmission data register L | RLN3nLUWTDRL | <RLIN3n_base> + 28 _H | — | √ |
| RLIN3n | UART wait transmission data register H | RLN3nLUWTDRH | <RLIN3n_base> + 29 _H | — | √ |

Note: √: Used, —: Not used
When writing to an unused register, write the value after reset.

16.3.2 LIN Master Related Registers

16.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------|-----|-----|-----|-----------|-----|-----|-------|
| | NSPB[3:0] | | | | LPRS[2:0] | | | LWBR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.12 RLN3nLWBR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 4 | NSPB[3:0] | Bit Sampling Count Select b_7 b_4 0 0 0 0: 16 sampling 1 1 1 1: 16 sampling Settings other than the above are prohibited. |
| 3 to 1 | LPRS[2:0] | Prescaler Clock Select b_3 b_1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128 |
| 0 | LWBR0 | Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified in the LCKS bit of the RLN3nLMD register is used. (LIN1.3) 1: In LIN wake-up mode, the clock f_a is used regardless of the setting in the LCKS bit of the RLN3nLMD register. (LIN2.x) |

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).

In LIN master mode (LIN/UART mode select bits in LIN mode register = 00_B), set these bits to 0000_B or 1111_B (16 sampling).

LPRS[2:0] Bits (Prescaler Clock Select)

These bits select the frequency division ratio for the prescaler.

The LIN communication clock source is divided by this prescaler.

LWBR0 Bit (Wake-up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects f_a as the LIN system clock (f_{LIN}) during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed). This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

Setting the baud rate to 19200 bps while f_a is selected allows an input signal with a low-level width of 130 μ s or longer to be detected during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit.

16.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read/written in 8-bit units

Address: <RLN3n_base> + 02_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LBRP0[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.13 RLN3nLBRP0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 7 to 0 | LBRP0[7:0] | Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting range: 00 _H to FF _H |

Set the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources f_a , f_b , and f_c .

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by N + 1.

The RLN3nLBRP0 and RLN3nLBRP1 registers are accessible in a 16-bit unit as RLN3nLBRP01.

16.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 03_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LBRP1[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.14 RLN3nLBRP1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 7 to 0 | LBRP1[7:0] | Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting range: 00 _H to FF _H |

Set the RLN3nLBRP1 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M+1.

The RLN3nLBRP0 and RLN3nLBRP1 registers are accessible in 16-bit units as RLN3nLBRP01.

16.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 04_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|-----|-----|-----|-----|-----|-----|-----|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | LSTM |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.15 RLN3nLSTC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 0 | — | Writing A7 _H , 58 _H , and 01 _H successively to the RLN3nLSTC register places the module into LIN self-test mode. |
| 0 | LSTM | LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode. |

The RLN3nLSTC register cancels protection of LIN self-test mode.

Set the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Writing A7_H, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed and the mode is changed to LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 16.9, LIN Self-Test Mode**.

When read, bits 6 to 1 return “000000_B”, and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.

For leaving LIN self-test mode, see **Section 16.9, LIN Self-Test Mode**.

Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of A7_H, 58_H, and 01_H.

16.3.2.5 RLN3nLMD — LIN Mode Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base>+ 08_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|--------|------|-----------|-----|----------|-----|
| | — | — | LRDNFS | LIOS | LCKS[1:0] | | LMD[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.16 RLN3nLMD Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | LRDNFS | LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled. |
| 4 | LIOS | LIN Interrupt Output Select 0: No RLIN3n interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used. |
| 3, 2 | LCKS[1:0] | LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1) |
| 1, 0 | LMD[1:0] | LIN/UART Mode Select b1 b0 0 0: LIN master mode |

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode)

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.

With 0 set, no RLIN3 interrupt is generated from the LIN/UART interface.

With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are generated from the LIN/UART interface.

For each interrupt source, see **Section 16.4, Interrupt Sources**.

LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.

With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).

With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).

With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).

With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x is used), and the RLN3nLMST register is 01_H (LIN wake-up mode), the protocol controller is supplied with fa regardless of the setting of this bit (the LCKS bit is not changed).

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.

To use the LIN/UART interface as an LIN master, set these bits to 00_B .

16.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|----------|-----|----------|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | BDT[1:0] | | BLT[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.17 RLN3nLBFC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5, 4 | BDT[1:0] | Transmission Break Delimiter (High Level) Width Select b_5 b_4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits |
| 3 to 0 | BLT[3:0] | Transmission Break (Low Level) Width Select b_3 b_0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits |

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header. 1 Tbit to 4 Tbits can be set.

BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header. 13 Tbits to 28 Tbits can be set.

16.3.2.7 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base>+ 0A_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|----------|-----|---|-----------|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | IBS[1:0] | | — | IBHS[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R/W | R/W | R/W |

Table 16.18 RLN3nLSC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5, 4 | IBS[1:0] | Inter-Byte Space Select b_5 b_4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits |
| 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | IBHS[2:0] | Inter-Byte Space (Header)/Response Space Select b_2 b_0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits |

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.

0 Tbit to 3 Tbits can be set.

These bits are enabled only during response transmission; these are disabled during response reception. For transmission from the transmission data buffer (RLN3nLUTDR register) or the transmission data buffer for wait (RLN3nLUWTDR register), the setting of these bits is ineffective. In this case, set these bits to “00_B”.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.

0 Tbit to 7 Tbits can be set.

The response space setting is enabled only during response transmission; setting is disabled during response reception.

The inter-byte space (header) is equal to the response space.

16.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base>+ 0B_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|-----------|-----|-----|-----|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WUTL[3:0] | | | | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R | R |

Table 16.19 RLN3nLWUP Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 4 | WUTL[3:0] | Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits |
| 3 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission. 1 Tbit to 16 Tbits can be set.

With 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x is used), fa is selected as the LIN system clock (f_{LIN}) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

16.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 0C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------|-------|-------|-------|
| | — | — | — | — | SHIE | ERRIE | FRCIE | FTCIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 16.20 RLN3nLIE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | SHIE | Successful Header Transmission Interrupt Enable 0: Disables successful header transmission interrupt. 1: Enables successful header transmission interrupt. |
| 2 | ERRIE | Error Detection Interrupt Enable 0: Disables error detection interrupt. 1: Enables error detection interrupt. |
| 1 | FRCIE | Successful Frame/Wake-up Reception Interrupt Enable 0: Disables successful frame/wake-up reception interrupt. 1: Enables successful frame/wake-up reception interrupt. |
| 0 | FTCIE | Successful Frame/Wake-up Transmission Interrupt Enable 0: Disables successful frame/wake-up transmission interrupt. 1: Enables successful frame/wake-up transmission interrupt. |

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

SHIE Bit (Successful Header Transmission Interrupt Enable)

The SHIE bit enables or disables interrupt request upon successful transmission of a header.

With 0 set, the interrupt request for RLIN3n transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.

With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.

Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors.

Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Frame/Wake-up Reception Interrupt Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (input signal low-level width count).

With 0 set, the interrupt request for RLIN3n reception complete is not generated when the FRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n reception complete is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Frame/Wake-up Transmission Interrupt Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a frame or a wake-up signal.

With 0 set, the interrupt request for RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

16.3.2.10 RLN3nLEDE — LIN Error Detection Enable Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 0D_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------|---|---|---|------|-------|-------|------|
| | LTES | — | — | — | FERE | FTERE | PBERE | BERE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R/W | R/W | R/W | R/W |

Table 16.21 RLN3nLEDE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | LTES | Timeout Error Select 0: Frame timeout error 1: Response timeout error |
| 6 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | FERE | Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection. |
| 2 | FTERE | Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection. |
| 1 | PBERE | Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection. |
| 0 | BERE | Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection. |

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.

With 0 set, the timeout function is used for frame timeout.

With 1 set, the timeout function is used for response timeout.

For details on the timeout error, see **Section 16.7.6, Error Status**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.

For details on the framing error, see **Section 16.7.6, Error Status**.

FTERE Bit (Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.

With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is indicated in the FTER flag in the RLIN3nLEST register.

With the LTES bit, either the frame timeout error or response timeout error can be selected.

Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.

For details on the timeout error, see **Section 16.7.6, Error Status**.

PBERE Bit (Physical Bus Error Detection Enable)

The PBERE bit enables or disables detection of the physical bus error.

With 0 set, the physical bus error is not detected.

With 1 set, the physical bus error is detected.

When this bit is set to 1, the detection result is indicated in the PBER flag in the RLIN3nLEST register.

For details on the physical bus error, see **Section 16.7.6, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is indicated in the BER flag in the RLIN3nLEST register.

For details on the bit error, see **Section 16.7.6, Error Status**.

16.3.2.11 RLN3nLCUC — LIN Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 0E_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | OM1 | OM0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 16.22 RLN3nLCUC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | OM1 | LIN Mode Select 0: LIN wake-up mode 1: LIN operation mode |
| 0 | OM0 | LIN Reset 0: LIN reset mode 1: Release from LIN reset mode |

Set the RLN3nLCUC register to 01_H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the register to 03_H to cause a transition to LIN operation mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.

Setting this bit to 0 causes a transition to LIN wake-up mode.

Setting it to 1 leads to release from LIN operation mode.

This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.

Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.

Setting this bit to 0 causes a transition to LIN reset mode.

Setting it to 1 leads to release from LIN reset mode.

16.3.2.12 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | RTS | FTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 16.23 RLN3nLTRC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | RTS | Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode. |
| 0 | FTS | Frame Transmission/Wake-up Transmission /Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission reception is started. |

RTS Bit (Response Transmission/Reception Start)

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 at the end of data group communication or transition to LIN reset mode.

FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)

Set the FTS bit to 1 to start frame transmission.

Also set this bit to 1 to allow wake-up transmission and reception (input signal low-level width count).

Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

16.3.2.13 RLN3nLMST — LIN Mode Status Register

Access: This register can only be read in 8-bit units

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|------|------|
| | — | — | — | — | — | — | OMM1 | OMM0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 16.24 RLN3nLMST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. |
| 1 | OMM1 | LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode. |
| 0 | OMM0 | LIN Reset Status Monitor 0: LIN reset mode. 1: LIN reset mode. |

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

16.3.2.14 RLN3nLST — LIN Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------|------|---|---|-----|---|-----|-----|
| | HTRC | D1RC | — | — | ERR | — | FRC | FTC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R/W | R/W |

Table 16.25 RLN3nLST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 | HTRC | Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed. |
| 6 | D1RC | Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed. |
| 5, 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | ERR | Error Detection Flag 0: No error has been detected. 1: Error has been detected. |
| 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | FRC | Frame/Wake-up Reception Complete Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed. |
| 0 | FTC | Frame/Wake-up Transmission Complete Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed. |

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt request for RLIN3n transmission is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC

register is 1), write 0 to the bit in LIN operation mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLIN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Frame/Wake-up Reception Complete Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for RLIN3n reception complete is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Frame/Wake-up Transmission Complete Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLIN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

16.3.2.15 RLN3nLEST — LIN Error Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------|---|------|---|-----|------|------|-----|
| | RPER | — | CSER | — | FER | FTER | PBER | BER |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R/W | R | R/W | R/W | R/W | R/W |

Table 16.26 RLN3nLEST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | RPER | Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected. |
| 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | CSER | Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected. |
| 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | FER | Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected. |
| 2 | FTER | Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected. |
| 1 | PBER | Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected. |
| 0 | BER | Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected. |

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the value of the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FTER Flag (Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the FTERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), the FTER flag is set to 1 upon frame timeout error or response timeout error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the PBERE bit of the RLN3nLEDE register is 1 (physical bus error detection enabled), the PBER flag is set to 1 upon physical bus error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

16.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|-----|-----|-----|-----------|-----|-----|-----|
| | LSS | FSM | CSM | RFT | RFDL[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.27 RLN3nLDFC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 | LSS | Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.) |
| 6 | FSM | Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set. |
| 5 | CSM | Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode |
| 4 | RFT | Response Field Communication Direction Select 0: Reception 1: Transmission |
| 3 to 0 | RFDL[3:0] | Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited. |

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received.

With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.

With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response communication mode.

With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN3nLTRC register is 1), response is transmitted/received without setting the RTS bit in the RLN3nLTRC register.

With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.

When causing a transition to LIN self-test mode, set this bit to 0 before transition.

For details on frame separate mode, see **Section 16.7.3.1, Transmission of LIN Frames**.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

CSM Bit (Checksum Select)

The CSM bit sets the checksum mode.

With 0 set, classic checksum mode is selected.

With 1 set, enhanced checksum mode is selected.

When the timeout error is used (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 16.7.6, Error Status**.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.

During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

RFT Bit (Response Field Communication Direction Select)

The RFT bit sets the direction of the response field/wake-up signal communication.

With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).

With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.

The data length can be 0 to 8 bytes excluding the checksum size.

To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).

To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the RTS bit in the RLN3nLTRC register is 0).

To receive response data, set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).

When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).

Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

16.3.2.17 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|------|------|---------|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IDP1 | IDP0 | ID[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.28 RLN3nLIDB Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | IDP1 | Parity Setting (P1) Sets the parity bits (P1) to be transmitted in the ID field. |
| 6 | IDP0 | Parity Setting (P0) Sets the parity bits (P0) to be transmitted in the ID field. |
| 5 to 0 | ID[5:0] | ID Setting Sets the 6-bit ID value to be transmitted in the ID field. |

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN self-test mode, this register operates as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 16.9, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 is for P0 and IDP1 for P1.

Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

16.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register can only be read in 8-bit units. In LIN self-test mode, this register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CKSM[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.29 RLN3nLCBR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 0 | CKSM[7:0] | Holds the checksum value transmitted or received. |

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
The value transmitted can be read. Read the value after transmission is completed.
Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
The value received can be read. Read the value after reception is completed.
Writing to this register is invalid.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 16.9, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for other data groups.

16.3.2.19 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read/written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
 RLN3nLDBR2: <RLIN3n_base> + 19_H
 RLN3nLDBR3: <RLIN3n_base> + 1A_H
 RLN3nLDBR4: <RLIN3n_base> + 1B_H
 RLN3nLDBR5: <RLIN3n_base> + 1C_H
 RLN3nLDBR6: <RLIN3n_base> + 1D_H
 RLN3nLDBR7: <RLIN3n_base> + 1E_H
 RLN3nLDBR8: <RLIN3n_base> + 1F_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LDB[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.30 RLN3nLDBRb Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 0 | LDB[7:0] | Sets the data to be transmitted or reads the received data. Setting range: 00 _H to FF _H |

- For response transmission:

The LDBRn registers set the data to be transmitted in the response field.
Use these registers with the following settings.

 - RFT in RLN3nLDFC register is 1 (transmission)
 - FSM in RLN3nLDFC register is 0 (not frame separate mode)
 - FTS in RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted)

or

 - RFT in RLN3nLDFC register is 1 (transmission)
 - FSM in RLN3nLDFC register is 1 (frame separate mode)
 - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For response reception:

The LDBRn registers hold the data received in the response field.
The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)
- For transmission of response data of 9 bytes or more:

Use the LDBRn registers with the following settings.

 - RFT in RLN3nLDFC register is 1 (transmission)
 - FSM in RLN3nLDFC register is 1 (frame separate mode)
 - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For reception of response data of 9 bytes or more:

Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 16.9, LIN Self-Test Mode**.

16.3.3 UART Related Registers

16.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|-----------|-----|-----|-----|-----------|-----|-----|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NSPB[3:0] | | | | LPRS[2:0] | | | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 16.31 RLN3nLWBR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 4 | NSPB[3:0] | Bit Sampling Count Select b_7 b_4 0 0 0 0: 16 sampling 0 1 0 1: 6 sampling 0 1 1 0: 7 sampling 0 1 1 1: 8 sampling 1 0 0 0: 9 sampling 1 0 0 1: 10 sampling 1 0 1 0: 11 sampling 1 0 1 1: 12 sampling 1 1 0 0: 13 sampling 1 1 0 1: 14 sampling 1 1 1 0: 15 sampling 1 1 1 1: 16 sampling Settings other than the above are prohibited. |
| 3 to 1 | LPRS[2:0] | Prescaler Clock Select b_3 b_1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128 |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Set the LN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate). In UART mode, it is possible to set the NSPB bits from 6 sampling to 16 sampling.

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler.

The LIN communication clock source is divided by this prescaler.

16.3.3.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 register can be read/written in 16-bit units.
 RLN3nLBRP0 register can be read/written in 8-bit units.
 RLN3nLBRP1 register can be read/written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H
 RLN3nLBRP0: <RLIN3n_base> + 02_H
 RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | BRP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.32 RLN3nLBRP01 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 15 to 0 | LBRP0[15:0] | Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000 _H to FFFF _H |

Set the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by the registers RLN3nLBRP0 and RLN3nLBRP1.

16.3.3.3 RLN3nLMD — UART Mode Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 08_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|--------|---|---|---|----------|-----|
| | — | — | LRDNFS | — | — | — | LMD[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R | R | R | R/W | R/W |

Table 16.33 RLN3nLMD Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | LRDNFS | UART Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled. |
| 4 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | LMD[1:0] | LIN/UART Mode Select b1 b0 0 1: UART mode |

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (UART Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.

To use the LIN/UART interface as an UART, set these bits to 01_B.

16.3.3.4 RLN3nLBFC — UART Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|------|------|----------|-----|-------|------|------|
| | — | UTPS | URPS | UPS[1:0] | | USBLS | UBOS | UBLS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.34 RLN3nLBFC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | UTPS | UART Output Polarity Switch 0: Transmit data normal output 1: Transmit data inverted output |
| 5 | URPS | UART Input Polarity Switch 0: Reception data normal output 1: Reception data inverted output |
| 4, 3 | UPS[1:0] | UART Parity Select 00: Parity disabled 01: Even parity 10: 0 Parity 11: Odd parity |
| 2 | USBLS | UART Stop Bit length Select 0: Stop bit: 1 bit 1: Stop bit: 2 bits |
| 1 | UBOS | UART Transfer Format Order Select 0: LSB First 1: MSB First |
| 0 | UBLS | UART Character Length Select 0: UART 8 bits communication 1: UART 7 bits communication |

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

UTPS Bit (UART Output Polarity Switch)

Sets the output polarity for UART communication.

With 0 set, transmit data is output without inversion.

With 1 set, inverted transmit data is output.

The setting of this bit is valid for all the bits of the UART frame.

In half-duplex communication, this setting should match the setting of URPS bit.

URPS Bit (UART Input Polarity Switch)

Sets the input polarity for UART communication.

With 0 set, receive data is input without inversion.

With 1 set, receive data is input with inversion.

The setting of this bit is valid for all the bits of the UART frame.

In half-duplex communication, this setting should match the setting of UTPS bit.

When setting this bit to “1” and expansion bit reception ((with expansion bit comparison) or (with data

comparison)) is performed, set the UEBDL bit in the RLN3nLUOR1 register and RLN3nLIDB register to the inverses of the expected values to enable comparison of the inverses of the received values.

UPS[1:0] Bits (UART Parity Select)

Sets the UART parity.

- When these bits are set to “00_B”, data is communicated without the parity.

[Transmission]

A parity bit is not added to transmit data.

[Reception]

Data is received without parity processing. Therefore, a parity error does not occur.

- When these bits are set to “01”, data is communicated with the even parity.

[Transmission]

If the number of 1s in transmit data is odd, “1” is added to the parity bit. If the number of 1s in transmit data is even, “0” is added to the parity bit.

[Reception]

If the number of 1s in receive data including the parity bit is odd, a parity error occurs.

- When these bits are set to “10”, data is communicated with 0 parity.

[Transmission]

Regardless of the number of 1s in transmit data, “0” is added to the parity bit.

[Reception]

The value of the parity bit is not judged. Therefore, no parity error occurs.

- When these bits are set to “11”, data is communicated with the odd parity.

[Transmission]

If the number of 1s in transmit data is odd, “0” is added to the parity bit. If the number of 1s in transmit data is even, “1” is added to the parity bit.

[Reception]

If the number of 1s in receive data including the parity bit is even, a parity error occurs.

USBLS Bit (UART Stop Bit Length Select)

Sets the stop bit length of data for UART communication.

With 0 set, stop bit length of 1 bit is selected.

With 1 set, stop bit length of 2 bits is selected.

UBOS Bit (UART Transfer Format Select)

Sets the bit order of data for UART communication.

With 0 set, LSB first is selected.

With 1 set, MSB first is selected.

UBLS Bit (UART Character Length Select)

Sets the character length of one frame for UART communication.

With 0 set, the character length is 8 bits.

With 1 set, the character length is 7 bits.

When the character length of one frame is 9 bits (the UEBE bit in the RLIN3nLUOR1 register is 1), the setting of this bit is ignored.

16.3.3.5 RLIN3nLSC — UART Space Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 0A_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|----------|-----|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | IBS[1:0] | | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R | R |

Table 16.35 RLIN3nLSC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5, 4 | IBS[1:0] | Inter-Byte Space Select $\begin{matrix} b5 & b4 \\ 0 & 0: 0 \text{ Tbit} \\ 0 & 1: 1 \text{ Tbit} \\ 1 & 0: 2 \text{ Tbits} \\ 1 & 1: 3 \text{ Tbits} \end{matrix}$ |
| 3 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Set the RLIN3nLSC register when the OMM0 bit in the RLIN3nLMST register is 0_B (LIN reset mode).

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the space between the UART frames when transmitting data from the UART buffer. 0 to 3 Tbits can be set.

16.3.3.6 RLN3nLEDE —UART Error Detection Enable Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 0D_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------|------|---|------|
| | — | — | — | — | FERE | OERE | — | BERE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R | R/W |

Table 16.36 RLN3nLEDE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | FERE | Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection. |
| 2 | OERE | Overrun Error Detection Enable 0: Disables overrun error detection. 1: Enables overrun error detection. |
| 1 | Reserved | When read, the value after reset is read. When writing to this bit, write the value after reset. |
| 0 | BERE | Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection. |

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.

For details on the framing error, see **Section 16.8.5, Error Status**.

OERE Bit (Overrun Error Detection Enable)

This bit enables or disables detection of the overrun error.

With 0 set, the overrun error is not detected.

With 1 set, the overrun error is detected.

When this bit is set to 1, the detection result is reflected in the OER flag in the RLN3nLEST register.

For details on the overrun error, see **Section 16.8.5, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is indicated in the BER flag in the RLIN3nLEST register.

In full-duplex communication, do not set this bit to “1”.

Do not set this register when the NSPB bits in the RLIN3nLWBR register are 0101_B (6 sampling) and the LRDNFS bit in the RLIN3nLMD register is 0 (noise filter is enabled).

For details on the bit error, see **Section 16.8.5, Error Status**.

16.3.3.7 RLIN3nLCUC — UART Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 0E_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----|
| | — | — | — | — | — | — | — | OM0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 16.37 RLIN3nLCUC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | OM0 | LIN Reset 0: LIN reset mode 1: Release from LIN reset mode |

After a value is written to this register, confirm that the value written is actually indicated in the RLIN3nLMST register before writing another value.

OM0 Bit (LIN Reset)

The OM0 bit selects either a transition to LIN reset mode or release from LIN reset mode.

Setting this bit to 0 causes a transition to LIN reset mode.

Setting it to 1 leads to release from LIN reset mode.

16.3.3.8 RLN3nLTRC — UART Transmission Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|-----|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | RTS | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R |

Table 16.38 RLN3nLTRC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | RTS | UART Buffer Transmission Start 0: UART Buffer transmission is stopped. 1: UART Buffer transmission is started. |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

RTS Bit (UART Buffer Transmission Start)

When transmitting data from the UART buffer, set this bit to “1”.

Only 1 can be written to this bit; 0 cannot be written.

Write to this bit when the UTOE bit in the RLN3nLUOER register is 1 (transmission enabled) and the UTS bit in the RLN3nLST register is 0 (transmission is not in progress).

Once set, regardless of errors, this bit is automatically cleared to 0 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register. This bit is also automatically cleared to 0 upon transition to LIN reset mode.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

When writing 1 to this bit while the UTSW bit in the RLN3nLRFC register is 1 (when UART buffer transmission is requested, the start of transmission is delayed until the stop bit of reception data is completed), write only during the reception of the stop bit.

16.3.3.9 RLIN3nLMST — UART Mode Status Register

Access: This register can only be read in 8-bit units

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | OMM0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 16.39 RLIN3nLMST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | OMM0 | LIN Reset Status Monitor 0: LIN reset mode 1: Not in LIN reset mode |

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

16.3.3.10 RLN3nLST — UART Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|-----|-----|-----|---|---|-----|
| | — | — | URS | UTS | ERR | — | — | FTC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 16.40 RLN3nLST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | URS | Data Reception Status 0: Reception is not operated. 1: Reception is operated. |
| 4 | UTS | Transmission Status 0: Transmission is not operated. 1: Transmission is operated. |
| 3 | ERR | Error Detection Flag 0: No error has been detected. 1: Error has been detected. |
| 2, 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | FTC | Successful UART Buffer Transmission Flag 0: UART buffer transmission has not been completed. 1: UART buffer transmission has been completed. |

The RLN3nLST register is automatically cleared to “00_H” upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains “00_H”. To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits by using the store instruction.

URS Bit (Data Reception Status)

At the start of the reception, this flag is set to 1.

The reception is started under the following condition.

- When the start bit is detected

At the end of reception, this flag is cleared to 0. While reception is stopped, this flag retains 0.

The reception is ended under the following condition.

- Sampling point of the first bit of the stop bits

UTS Bit (Transmission Status Flag)

At the start of the transmission, this flag is set to 1. During the transmission, this flag retains 1.

The transmission is started under the following conditions.

- When transmission data is set to the RLN3nLUTDR or RLN3nLUWTDR register
- When the RTS bit in the RLN3nLTRC register is set to 1

This flag is cleared to 0 at the end of transmission.

The transmission is ended under the following conditions.

- When transmission of data set in the RLN3nLUTDR or RLN3nLUWTDR register is completed and next data is not set
- When transmission from UART buffer is completed (when the RTS bit in the RLN3nLTRC register is cleared to 0)

ERR Flag (Error Detection Flag)

This flag is set to 1 upon detection of an error, detection of an expansion bit, or upon ID matching (when at least one of the flags of the RLN3nLEST register is 1). At this time, an interrupt request for RLIN3n status is generated. However, when this bit is 1, an interrupt is not generated upon detection of an error, detection of an expansion bit, or upon ID matching. To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register.

FTC Flag (Successful UART Buffer Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that it was before 1 was written.

Regardless of errors, this bit is set to 1 upon completion of transmission of the number of data specified by the MDL bit in the RLN3nLDFC register from the UART buffer. At this time, an interrupt request for RLIN3n transmission is generated. Write 0 to clear this flag.

16.3.3.11 RLN3nLEST — UART Error Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|------|------|------|-----|-----|---|-----|
| | — | UPER | IDMT | EXBT | FER | OER | — | BER |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R | R/W |

Table 16.41 RLN3nLEST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | UPER | Parity Error Flag 0: Parity error has not been detected. 1: Parity error has been detected. |
| 5 | IDMT | ID Matching Flag 0: The receive data does not match with the ID value. 1: The receive data matches with the ID value. |
| 4 | EXBT | Expansion Bit Detection Flag 0: Expansion bit has not been detected. 1: Expansion bit has been detected. |
| 3 | FER | Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected. |
| 2 | OER | Overrun Error Flag 0: Overrun error has not been detected. 1: Overrun error has been detected. |
| 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | BER | Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected. |

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to, and the value of 00_H is held. To clear certain bits in this register, write 0 to those bits, and write 1 to the bits not to be cleared by using the store instruction.

UPER Flag (Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

This flag is set to 1 upon parity error detection. Write 0 to clear this flag.

IDMT Flag (ID Matching Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

The IDMT flag becomes 1 when all the following conditions are met:

- The UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enabled)
- The UECD bit in the RLN3nLUOR1 register is 0 (expansion bit comparison enabled)

- The UEBDCE bit in the RLN3nLUOR1 register is 1 (data comparison after expansion bit is detected)
 - The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register match.
 - The value of the 8-bit of the received data excluding the expansion bit and the value of the RLN3nLIDB register match.

Write 0 to clear this flag.

EXBT Flag (Expansion Bit Detection Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

When the UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1.

Write 0 to clear this flag.

FER Flag (Framing Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

The FER flag is set to 1 upon framing error detection while the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled). Write 0 to clear this flag.

OER Flag (Overrun Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

The OER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). Write 0 to clear this flag.

BER Flag (Bit Error Flag)

Only 0 can be written to the this flag; when 1 is written, the bit retains the value that it was before 1 was written.

The BER flag is set to 1 when the transmitted data and the data monitored by the receive pin do not match while the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled).

Write 0 to clear this flag.

16.3.3.12 RLIN3nLDFC — UART Data Field Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|------|---|----------|-----|-----|-----|
| | — | — | UTSW | — | MDL[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R | R/W | R/W | R/W | R/W |

Table 16.42 RLIN3nLDFC Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|----------|--|----|----|--|---------|--|--------|---------|--|--------|---------|--|--------|---------|--|--------|---------|--|--------|---------|--|--------|---------|--|--------|---------|--|--------|---------|--|--------|---------|--|--------|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | UTSW | Transmission Start Wait 0: When UART buffer transmission is requested, transmission is started immediately. 1: When UART buffer transmission is requested, transmission is not started until reception of the stop bit is completed. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 to 0 | MDL[3:0] | UART Buffer Data Length Select <table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td></td> <td>9 data</td> </tr> <tr> <td>0 0 0 1</td> <td></td> <td>1 data</td> </tr> <tr> <td>0 0 1 0</td> <td></td> <td>2 data</td> </tr> <tr> <td>0 0 1 1</td> <td></td> <td>3 data</td> </tr> <tr> <td>0 1 0 0</td> <td></td> <td>4 data</td> </tr> <tr> <td>0 1 0 1</td> <td></td> <td>5 data</td> </tr> <tr> <td>0 1 1 0</td> <td></td> <td>6 data</td> </tr> <tr> <td>0 1 1 1</td> <td></td> <td>7 data</td> </tr> <tr> <td>1 0 0 0</td> <td></td> <td>8 data</td> </tr> <tr> <td>1 0 0 1</td> <td></td> <td>9 data</td> </tr> </table> Settings other than the above are prohibited. | b3 | b0 | | 0 0 0 0 | | 9 data | 0 0 0 1 | | 1 data | 0 0 1 0 | | 2 data | 0 0 1 1 | | 3 data | 0 1 0 0 | | 4 data | 0 1 0 1 | | 5 data | 0 1 1 0 | | 6 data | 0 1 1 1 | | 7 data | 1 0 0 0 | | 8 data | 1 0 0 1 | | 9 data |
| b3 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 0 0 | | 9 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 0 1 | | 1 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 1 0 | | 2 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 1 1 | | 3 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 0 0 | | 4 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 0 1 | | 5 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 1 0 | | 6 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 1 1 | | 7 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 0 0 | | 8 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 0 1 | | 9 data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

UTSW Bit (Transmission Start Wait)

This bit controls the transmission start timing of UART buffer.

With 0 set, transmission is started as soon as the start of UART buffer transmission is requested.

With 1 set, transmission is started after the completion of the stop bit reception.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLIN3nLBFC register.

This bit is enabled when the RTS bit in the RLIN3nLTRC register is set to 1. In addition, writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission started).

Set this bit to 1 only to switch from reception to transmission in half-duplex communication.

MDL[3:0] Bits (UART Buffer Data Length Select)

These bits specify the data length of the UART buffer.

Writing a value to these bits is disabled when the RTS bit in the RLIN3nLTRC register is 1 (UART buffer transmission started).

16.3.3.13 RLN3nLIDB — UART ID Buffer Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ID[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.43 RLN3nLIDB Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 0 | ID[7:0] | ID value that is referred for the expansion bit data comparison is set |

ID Bits (ID Bits)

When the UEBE bit in the RLN3nLUOR1 register is set to 1 (expansion bit enabled), the UECD bit is set to 0 (expansion bit comparison enabled), and the UEBDCE bit is set to 1 (data comparison after expansion bit is detected), set the value to be compared with the received data. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (receive operation is not in progress).

16.3.3.14 RLN3nLUDB0 — UART Data Buffer 0 Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 17_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | UDB[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.44 RLN3nLUDB0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---------------------------------------|
| 7 to 0 | UDB[7:0] | Sets the data to be UART transmitted. |

When transmitting 9-byte data from the UART buffer (the RLN3nLDFC.MDL bit is 0_H or 9_H), set the first data to be transmitted.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 16.45, Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 16.8.1.2, UART Buffer Transmission, (1) UART Buffer Transmission**.

Table 16.45 Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format

| | RLN3nLUDB0 | | | | | | | |
|------------------|------------|-------|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 7-bit; LSB first | – | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7-bit; MSB first | – | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 |
| 8-bit; LSB first | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 8-bit; MSB first | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |

16.3.3.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)

Access: This register can be read/written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
 RLN3nLDBR2: <RLIN3n_base> + 19_H
 RLN3nLDBR3: <RLIN3n_base> + 1A_H
 RLN3nLDBR4: <RLIN3n_base> + 1B_H
 RLN3nLDBR5: <RLIN3n_base> + 1C_H
 RLN3nLDBR6: <RLIN3n_base> + 1D_H
 RLN3nLDBR7: <RLIN3n_base> + 1E_H
 RLN3nLDBR8: <RLIN3n_base> + 1F_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|-----|-----|-----|-----|-----|-----|-----|
| | LDB[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.46 RLN3nLDBRb Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|----------------------------------|
| 7 to 0 | LDB[7:0] | Sets the data to be transmitted. |

This register specifies the data transmitted from the UART buffer.

Write to these registers when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 16.47, Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 16.8.1.2, UART Buffer Transmission, (1) UART Buffer Transmission**.

Table 16.47 Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format

| | RLN3nLDBRb | | | | | | | |
|------------------|------------|-------|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 7-bit; LSB first | – | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7-bit; MSB first | – | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 |
| 8-bit; LSB first | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 8-bit; MSB first | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |

16.3.3.16 RLN3nLUOER — UART Operation Enable Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 20_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|------|------|
| | — | — | — | — | — | — | UROE | UTOE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 16.48 RLN3nLUOER Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | UROE | Reception Enable 0: Disables reception. 1: Enables reception. |
| 0 | UTOE | Transmission Enable 0: Disables transmission. 1: Enables transmission. |

The RLN3nLUOER register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

UROE Bit (Reception Enable)

The UROE bit enables or disables reception.

With 0 set, reception is disabled.

With 1 set, reception is enabled.

Do not clear this bit during reception. If the communication is suspended during reception, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to shift to the LIN reset mode. However, the transmit operation is also suspended at this time.

When the UART buffer transmits data, do not set this bit to 1.

UTOE Bit (Transmission Enable)

The UTOE bit enables or disables transmission.

With 0 set, transmission is disabled.

With 1 set, transmission is enabled.

Do not clear this bit during transmission. If the communication is suspended during transmission, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to shift to the LIN reset mode.

However, the receive operation is also suspended at this time.

16.3.3.17 RLN3nLUOR1 — UART Option Register 1

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 21_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|------|--------|--------|-------|------|
| | — | — | — | UECD | UTIGTS | UEBDCE | UEBDL | UEBE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 16.49 RLN3nLUOR1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | UECD | Expansion Bit Comparison Disable 0: Enables expansion bit comparison. 1: Disables expansion bit comparison. |
| 3 | UTIGTS | Transmission Interrupt Generation Timing Select 0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission. |
| 2 | UEBDCE | Expansion Bit Data Comparison Enable 0: Disables data comparison after an expansion bit is detected. 1: Enables data comparison after an expansion bit is detected. |
| 1 | UEBDL | Expansion Bit Detection Level Select 0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level. |
| 0 | UEBE | Expansion Bit Enable 0: Disables expansion bit operation. 1: Enables expansion bit operation. |

UECD Bit (Expansion Bit Comparison Disable)

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).

With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.

With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

Do not set this bit to 1 when the UEBDCE bit is 1 (data comparison after expansion bit is detected).

UTIGTS Bit (Transmission Interrupt Generation Timing Select)

The UTIGTS bit sets the generation timing of the transmission interrupt.

With 0 set, the transmission interrupt is generated at the start of transmission.

With 1 set, the transmission interrupt is generated at the completion of transmission.

When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.

When transmission from the UART buffer is performed with 1 set, the transmission interrupt is

generated only at the completion of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.

UEBDCE Bit (Expansion Bit Data Comparison Enable)

After an expansion bit is detected, this bit enables or disables the comparison between the 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register.

With 0 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is disabled.

With 1 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is enabled.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Do not set this bit to 1 when the UEBE bit is 0 (expansion bit operation disabled).

Do not set this bit to 1 when the UECD bit is 1 (expansion bit comparison disabled).

Do not set this bit to 1 when the UART buffer is used.

UEBDL Bit (Expansion Bit Detection Level Select)

The UEBDL bit sets the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (comparison of the expansion bit is enabled).

With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.

With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

UEBE Bit (Expansion Bit Enable Bit)

The UEBE bit enables or disables expansion bit operation.

With 0 set, expansion bit operation is disabled.

With 1 set, expansion bit operation is enabled.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

16.3.3.18 RLN3nLUTDR — UART Transmission Data Register

Access: RLN3nLUTDR register can be read/written in 16-bit units.
RLN3nLUTDRL register can be read/written in 8-bit units.
RLN3nLUTDRH register can be read/written in 8-bit units.

Address: RLN3nLUTDR: <RLIN3n_base> + 24_H
RLN3nLUTDRL: <RLIN3n_base> + 24_H
RLN3nLUTDRH: <RLIN3n_base> + 25_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|----------|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | UTD[8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.50 RLN3nLUTDR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 to 0 | UTD[8:0] | Sets the data to be transmitted. Setting range: 000 _H to 1FF _H |

The RLN3nLUTDR register sets the data to be transmitted from the transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

This register can be accessed in 8 bits.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

Also, do not write data to this register when a transmission request is being generated due to write access to the RLN3nLUWTDR register.

When transmitting data continuously, do not set another piece of transmission data in this register before the generation of transmission interrupt.

The table below shows the bit arrangement according to the set communication format.

Table 16.51 Bit Arrangement of the RLN3nLUTDR Register According to Each Communication Format

| | RLN3nLUTDR | | | | | | | | |
|------------------|------------|------|------|------|------|------|------|------|------|
| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 7-bit; LSB first | — | — | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 7-bit; MSB first | — | — | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 |
| 8-bit; LSB first | — | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 8-bit; MSB first | — | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 |
| 9-bit; LSB first | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 9-bit; MSB first | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 |

16.3.3.19 RLN3nLURDR — UART Reception Data Register

Access: RLN3nLURDR register can only be read in 16-bit units.
RLN3nLURDRL register can only be read in 8-bit units.
RLN3nLURDRH register can only be read in 8-bit units.

Address: RLN3nLURDR: <RLIN3n_base> + 26_H
RLN3nLURDRL: <RLIN3n_base> + 26_H
RLN3nLURDRH: <RLIN3n_base> + 27_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|-----------|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | URD [8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 16.52 RLN3nLURDR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 15 to 9 | Reserved | When read, the value after reset is read. |
| 8 to 0 | URD [8:0] | Stores the received data Storing range: 000 _H to 1FF _H |

The RLN3nLURDR allows the reception data to be read from the receive data register.

When the UROE bit in the RLN3nLUOER register is 1, the reception data is stored in this register and can be read out.

This register is updated when the stop bit of the receive data is received.

This register is also updated when an error is caused by the parity or stop bit.

However, the value of this register is not updated upon occurrence of an overrun error when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). The value of this register is updated upon occurrence of an overrun error when the OERE bit is 0 (overrun error detection disabled).

Read this register upon occurrence of a receive error (overrun error, framing error, parity error) when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). If the next data is received without reading this register, an overrun error occurs.

This register can be accessed in 8 bits.

However, during expansion bit use (UEBE bit of the RLN3nLUOR1 register is 1 (expansion bit operation enabled)), do not attempt 8-bit access.

The table below shows the bit arrangement according to the set communication format.

Table 16.53 Bit Arrangement of the RLN3nLURDR Register According to Each Communication Format

| | RLN3nLURDR | | | | | | | | |
|------------------|------------|------|------|------|------|------|------|------|------|
| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 7-bit; LSB first | — | — | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 7-bit; MSB first | — | — | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 |
| 8-bit; LSB first | — | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 8-bit; MSB first | — | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 |
| 9-bit; LSB first | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 9-bit; MSB first | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 |

16.3.3.20 RLN3nLUWTDR — UART Wait Transmission Data Register

Access: RLN3nLUWTDR register can be read/written in 16-bit units.
RLN3nLUWTDRL register can be read/written in 8-bit units.
RLN3nLUWTDRLH register can be read/written in 8-bit units.

Address: RLN3nLUWTDR: <RLIN3n_base> + 28_H
RLN3nLUWTDRL: <RLIN3n_base> + 28_H
RLN3nLUWTDRLH: <RLIN3n_base> + 29_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|-----------|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | UWTD[8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 16.54 RLN3nLUWTDR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 to 0 | UWTD[8:0] | Sets the data to be transmitted from the wait transmit data register after waiting for the stop bit reception to be completed. Setting range: 000 _H to 1FF _H |

The RLN3nLUWTDR register sets the data to be transmitted from the UART wait transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

Use this register only to switch from reception to transmission in half-duplex communication.

The user should write to this register only while the stop bit is being received.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

When this register is read, the RLN3nLUTDR register value is actually read.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set communication format.

Table 16.55 Bit Arrangement of the RLN3nLUWTDR Register According to Each Communication Format

| | RLN3nLUWTDR | | | | | | | | |
|------------------|-------------|------|------|------|------|------|------|------|------|
| | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 7-bit; LSB first | — | — | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 7-bit; MSB first | — | — | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 |
| 8-bit; LSB first | — | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 8-bit; MSB first | — | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 |
| 9-bit; LSB first | Bit8 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 9-bit; MSB first | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 |

16.4 Interrupt Sources

The LIN/UART interface generates three types of interrupt requests.

- RLIN3n successful transmission interrupt
- RLIN3n successful reception interrupt
- RLIN3n status interrupt

When the LIOS bit in the RLN3nLMD register is set to 0, no interrupt occurs.

Setting the LIOS bit in the RLN3nLMD register to 1 allows to output the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, or RLIN3n status interrupt depending on the interrupt source.

Table 16.56 lists the sources for each interrupt.

Table 16.56 Interrupt Sources

| | | LIOS bit in RLN3nLMD register is 1*1 | | |
|-----------|-----------------|--|--|--|
| | | RLIN3n Transmission Interrupt | RLIN3n Successful Reception Interrupt | RLIN3n Status Interrupt |
| LIN mode | LIN master mode | <ul style="list-style-type: none"> • Successful frame transmission • Successful wake-up transmission • Successful header transmission | <ul style="list-style-type: none"> • Successful wake-up reception • Successful wake-up reception | <ul style="list-style-type: none"> • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error |
| UART mode | | <ul style="list-style-type: none"> • Transmission start/successful transmission | <ul style="list-style-type: none"> • Successful reception • Expansion bit mismatch | <ul style="list-style-type: none"> • Bit error • Overrun error • Framing error • Expansion bit match • ID match • Parity error |

Note 1. The LIOS bit setting is valid in LIN Mode. In UART mode, setting the LIOS bit is not required.

In LIN mode, each interrupt request is output when the corresponding bit in the RLN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLN3nLST register is 1.

16.5 Modes

The LIN/UART interface provides the following four modes, depending upon the specific function to be performed:

- LIN reset mode
- LIN mode
 - LIN master mode
- UART mode
- LIN self-test mode

In LIN reset mode, clock supply to the LIN/UART interface stops. This enables to reduce power consumption.

Figure 16.2 shows mode transitions. **Table 16.57** describes mode transition conditions. **Table 16.58** lists operations available in each mode.

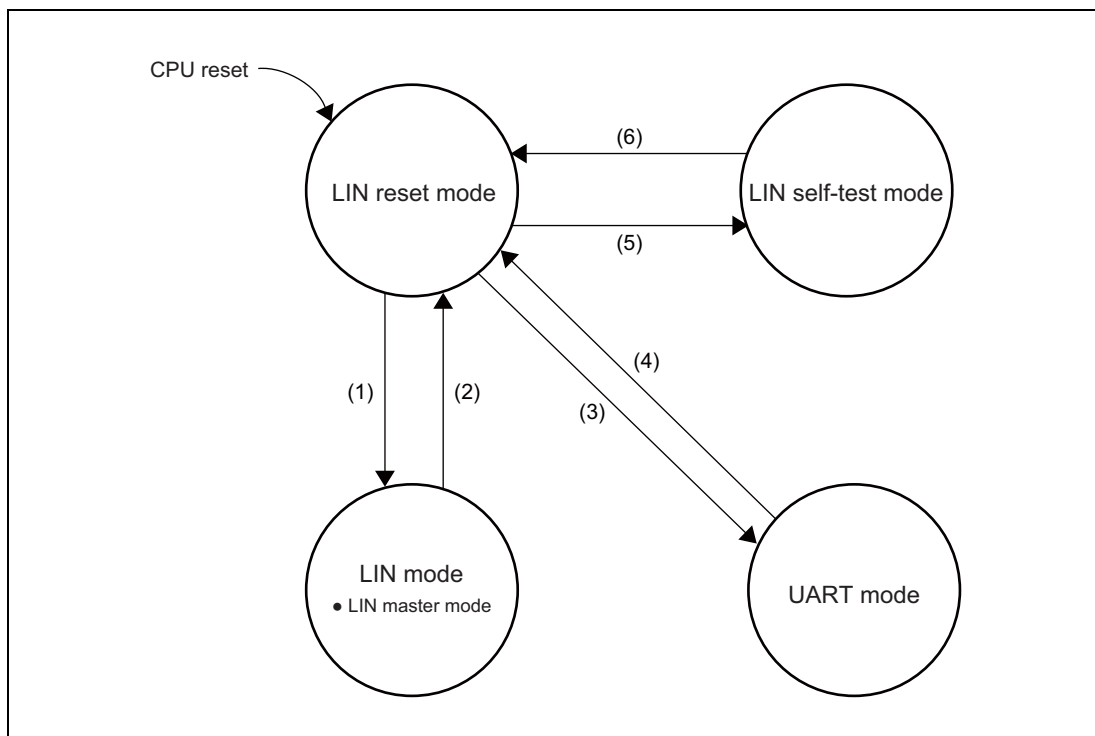


Figure 16.2 Mode Transitions

Table 16.57 Transition Condition of Each Mode

| Transition Mode | | Transition Condition |
|------------------------|----------------------|---|
| (1) LIN reset mode | → LIN mode | <ul style="list-style-type: none"> LIN master mode LMD bits in the RLN3nLMD register = 00_B and OM1 and OM0 bits in the RLN3nLCUC register = 01_B or 11_B |
| (2) LIN mode | → LIN reset mode | OM0 bit in RLN3nLCUC register = 0 _B |
| (3) LIN reset mode | → UART mode | LMD bits in RLN3nLMD register = 01 _B and OM0 bit in RLN3nLCUC register = 1 _B |
| (4) UART mode | → LIN reset mode | OM0 bit in RLN3nLCUC register = 0 _B |
| (5) LIN reset mode | → LIN self-test mode | See Section 16.9, LIN Self-Test Mode. |
| (6) LIN self-test mode | → LIN reset mode | See Section 16.9, LIN Self-Test Mode. |

Table 16.58 Operations Available in Each Mode

| LIN Mode | | |
|--|--|--------------------|
| LIN Master Mode | UART Mode | LIN Self-Test Mode |
| Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection | UART transmission UART reception Error detection | Self test |

Whether mode has transitioned to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the RLN3nLMD register and the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see **Section 16.9, LIN Self-Test Mode.**

16.6 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the RLN3nLMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication and the UART communication functions are halted.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

When the mode changes to LIN reset mode, the following registers are initialized to their reset values, and as long as LIN reset mode is in effect, they retain their initial values.

- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register
- RLN3nLUOER register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLUDB0 register
- RLN3nLDBRb register (b = 1 to 8)
- RLN3nLUOR1 register
- RLN3nLUTDR register
- RLN3nLURDR register
- RLN3nLUWTDR register

16.7 LIN Mode

LIN mode can operate in LIN master mode.

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 00_B (LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either 01_B or 11_B sets LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either 01_B to 11_B .

When changing a submode to another submode within LIN mode, a transition to LIN reset mode should first be made and change the LMD bits in the RLN3nLMD register.

The LIN mode provides the following two operation modes:

- LIN operation mode
- LIN wake-up mode

Figure 16.3 shows the transition of operation modes. **Table 16.59** describes the transition conditions of operation modes.

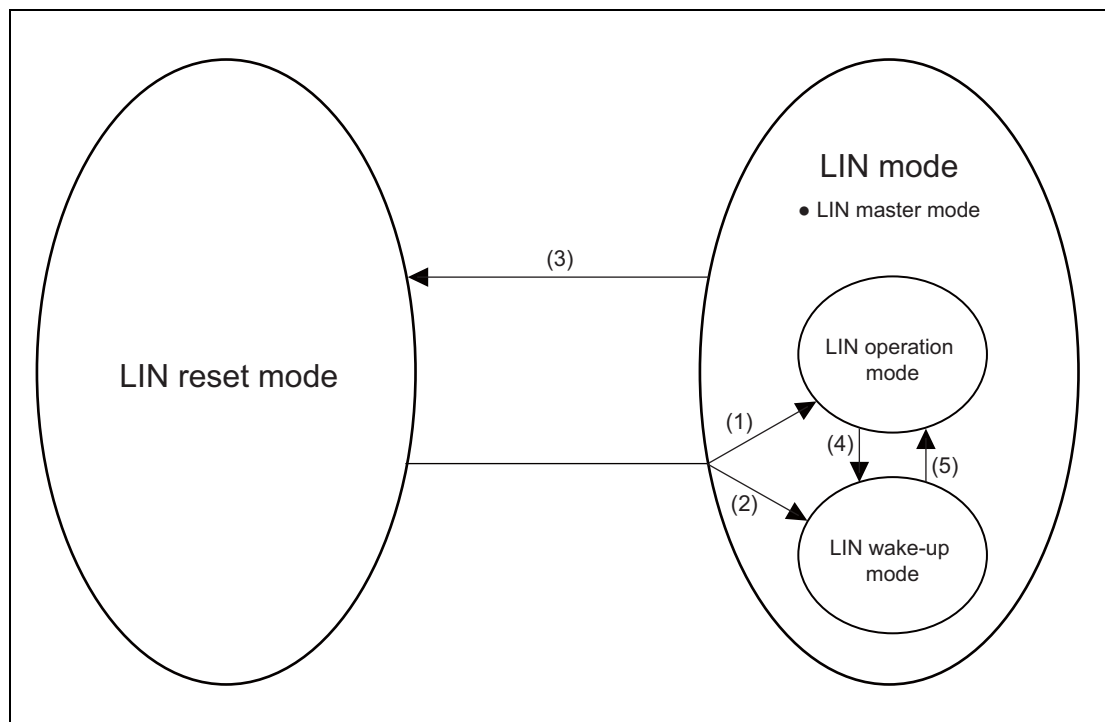


Figure 16.3 Transition of Operation Modes

Table 16.59 Transition Condition for Operation Mode

| Operation Mode Transition | | Transition Condition |
|---------------------------|---|--|
| (1) | LIN reset mode → LIN mode • LIN operation mode | LMD bits in RLN3nLMD register = 00 _B and OM1 and OM0 bits in RLN3nLCUC register = 11 _B |
| (2) | LIN reset mode → LIN mode • LIN wake-up mode | LMD bits in RLN3nLMD register = 00 _B and OM1 and OM0 bits in RLN3nLCUC register = 01 _B |
| (3) | LIN mode → LIN reset mode • LIN operation mode • LIN wake-up mode | OM0 bit in RLN3nLCUC register = 0 _B |
| (4) *1 | LIN mode → LIN mode • LIN operation mode • LIN wake-up mode | OM1 and OM0 bits in RLN3nLCUC register = 01 _B |
| (5) *1 | LIN mode → LIN mode • LIN wake-up mode • LIN operation mode | OM1 and OM0 bits in RLN3nLCUC register = 11 _B |

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is in progress (when the FTS bit in the RLN3nLTRC register is 1).

(1) LIN Operation Mode

In LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 11_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 11_B.

(2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 01_B.

16.7.1 LIN Master Mode

16.7.1.1 Header Transmission

Figure 16.4 shows the operation of the LIN/UART interface (LIN master mode) in header transmission. **Table 16.60** describes the processing in header transmission.

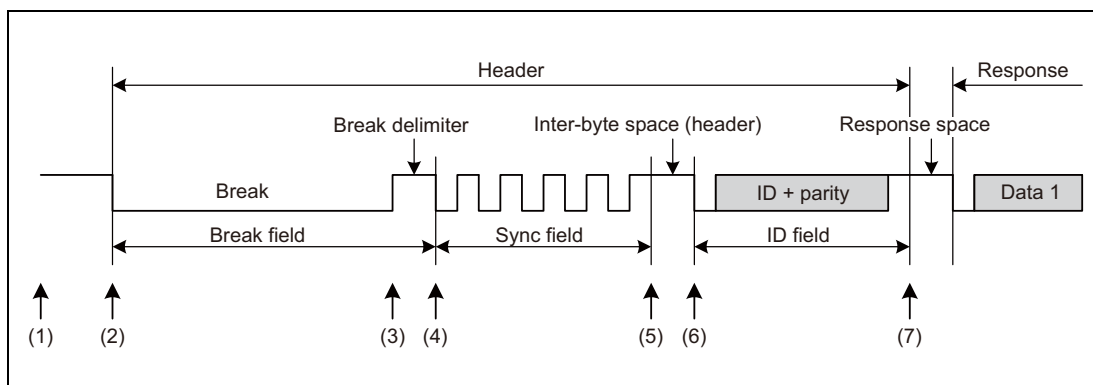


Figure 16.4 Operation in Header Transmission

Table 16.60 Processing in Header Transmission

| Software Processing | LIN/UART Interface Processing |
|---|--|
| (1) <ul style="list-style-type: none"> • Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN master mode: LIN operation mode • Sets information on the frame to be transmitted (ID, parity, data length, response direction, checksum method, and transmission data) | Waits for the setting of the FTS bit in the RLIN3nLTRC register by software (idle) |
| (2) Sets the FTS bit in the RLIN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started) | Transmits a break. |
| (3) Waits for an interrupt request | Transmits a break delimiter. |
| (4) | Transmits a sync field (55 _H). |
| (5) | Transmits an inter-byte space (header). |
| (6) | Transmits an ID field. |
| (7) | Sets a successful header transmission flag. |

NOTE

For information about error detection conditions, see **Section 16.7.6, Error Status**.

16.7.1.2 Response Transmission

Figure 16.5 shows the operation of the LIN/UART interface (LIN master mode) in response transmission. Table 16.61 describes the processing in response transmission.

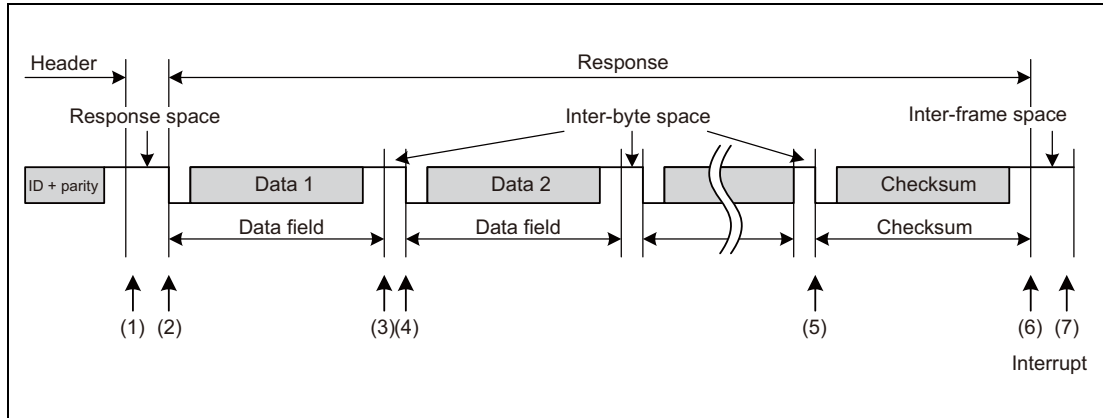


Figure 16.5 Operation in Response Transmission

Table 16.61 Processing in Response Transmission

| Software Processing | LIN/UART Interface Processing |
|---|---|
| (1) [When in frame separate mode] <ul style="list-style-type: none"> Sets the RTS bit in the RLIN3nLTRC register to 1 (response transmission/reception started) [When not in frame separate mode] <ul style="list-style-type: none"> Waits for an interrupt request | [When in frame separate mode] <ul style="list-style-type: none"> Waits for the setting of the RTS bit in the RLIN3nLTRC register to 1 by software. When the bit is set to 1, sends a response space. [When not in frame separate mode] <ul style="list-style-type: none"> Sends a response space. |
| (2) Waits for an interrupt request | Transmits data 1. |
| (3) | Transmits an inter-byte space. |
| (4) | <ul style="list-style-type: none"> Transmits data 2. Transmits an inter-byte space Transmits data 3. Transmits an inter-byte space (The transmission of an inter-byte space is repeated as many times as the data length specified in bits RFDL[3:0] in the RFC register. This is halted if the BER flag in the RLIN3nLEST register is 1 (detection of a bit error). Process (5), checksum transmission, does not proceed if an error occurs). <p style="text-align: center;">⋮</p> |
| (5) | Transmits the checksum. |
| (6) | <ul style="list-style-type: none"> Sets a successful frame/wake-up transmission flag. Sets the FTS bit in the RLIN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped) [When in frame separate mode] <ul style="list-style-type: none"> Set the RTS bit in the RLIN3nLTRC register to 0 (response transmission/reception stopped). |
| (7) <ul style="list-style-type: none"> Processing after communication Checks the RLIN3nLST register, and clears flags. | Idle |

NOTE

For information about error detection conditions, see Section 16.7.6, Error Status.

16.7.1.3 Response Reception

Figure 16.6 shows the operation of the LIN/UART interface (LIN master mode) in response reception. Table 16.62 describes the processing in response reception.

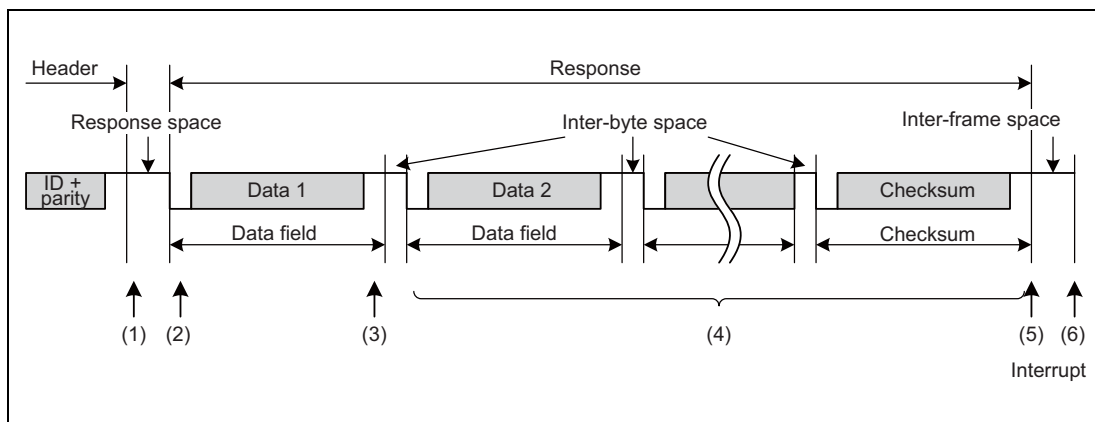


Figure 16.6 Operation in Response Reception

Table 16.62 Processing in Response Reception

| Software Processing | LIN/UART Interface Processing |
|---|--|
| (1) Waits for an interrupt request (no processing) | Waits for detection of a start bit. |
| (2) | Receives data 1 when the start bit is detected. |
| (3) | Sets the successful data 1 reception flag. |
| (4) | <ul style="list-style-type: none"> Receives data 2 when the start bit is detected. Receives data 3 when the start bit is detected. <p>(The transmission of an inter-byte space is repeated as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register. This is halted if any bit in the RLN3nLEST register is 1 (detection of any error). Process (5), checksum transmission, does not proceed if an error occurs).</p> <p style="text-align: center;">⋮</p> |
| (5) | <ul style="list-style-type: none"> Receives the checksum when the start bit is detected. Determines the checksum. Sets the successful frame/wake-up reception flag. Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped). |
| (6) <ul style="list-style-type: none"> Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags. | Idle |

NOTE

For information about error detection, see Section 16.7.6, Error Status.

16.7.2 Data Transmission/Reception

16.7.2.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag in the RLIN3nLEST register (see **Section 16.7.6, Error Status**).

In LIN master mode, 1 Tbit is generated to be $16/f_{LIN}$, and thus the sampling point for received data is at the 13th clock cycle (81.25% position).

Figure 16.7 shows an example of data transmission timing.

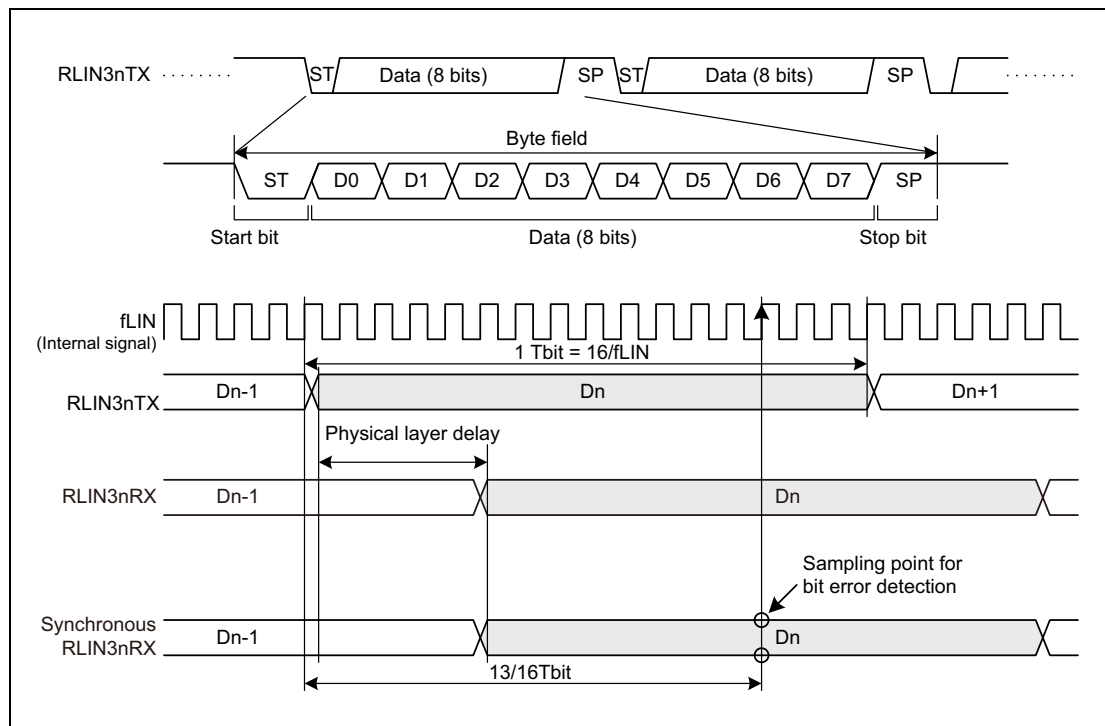


Figure 16.7 Example of Data Transmission Timing (LIN Master Mode)

16.7.2.2 Data Reception

Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN/UART interface has a noise filter function for reception data. If the LRDNFS bit in the RLN3nLMD register is 0, the LIN/UART interface uses a noise filter, and the value determined by a 3-sampling majority rule on prescaler clocks is used as the sampling value. If the LRDNFS bit in the RLN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and the value of the synchronized RLIN3nRX value at the sampling position is used as the sampling value.

Figure 16.8 shows an example of data reception timing.

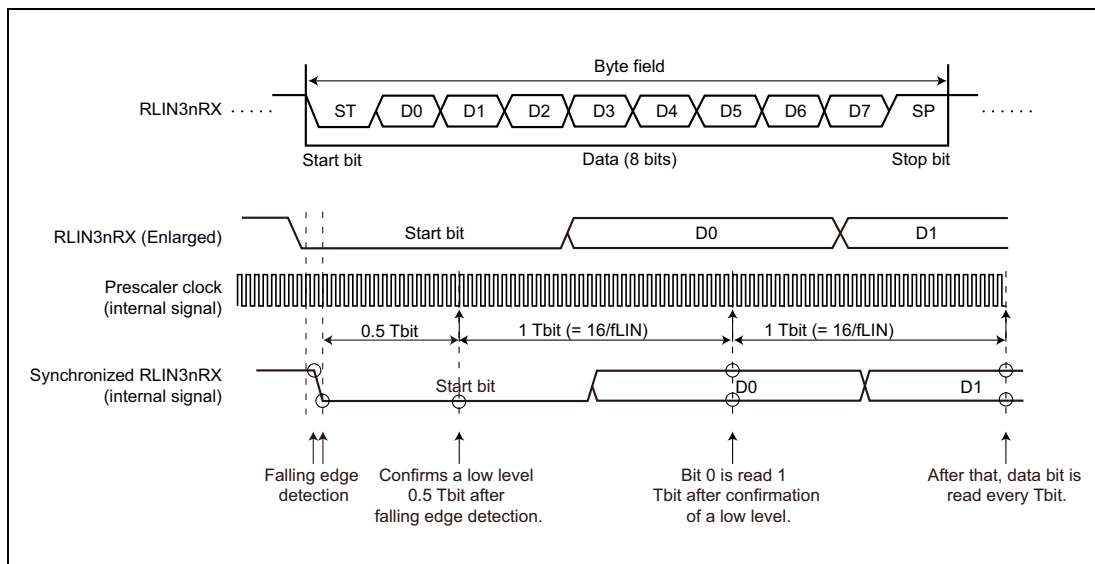


Figure 16.8 Example of Data Reception Timing (LIN Master Mode)

16.7.3 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface sends or receives data continuously.

16.7.3.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

Figure 16.9 shows the LIN transmission processing and the required buffers.

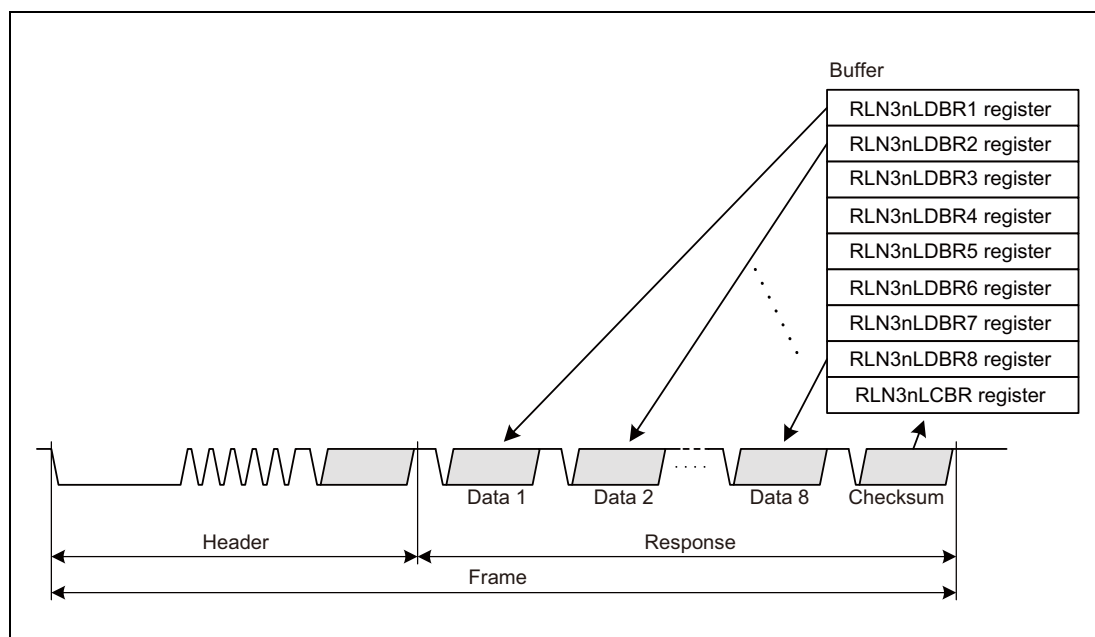


Figure 16.9 LIN Transmission Processing and Required Buffer

[Frame Separate Mode]

Setting the FSM bit in the RLN3nLDFC register to 1 turns on the frame separate mode.

In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN3nLST register is set to 1 (successful header transmission).

Use frame separate mode when transmitting or receiving response data of 9 bytes or more in LIN master mode.

16.7.3.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 16.10 shows the LIN reception processing and the required buffers.

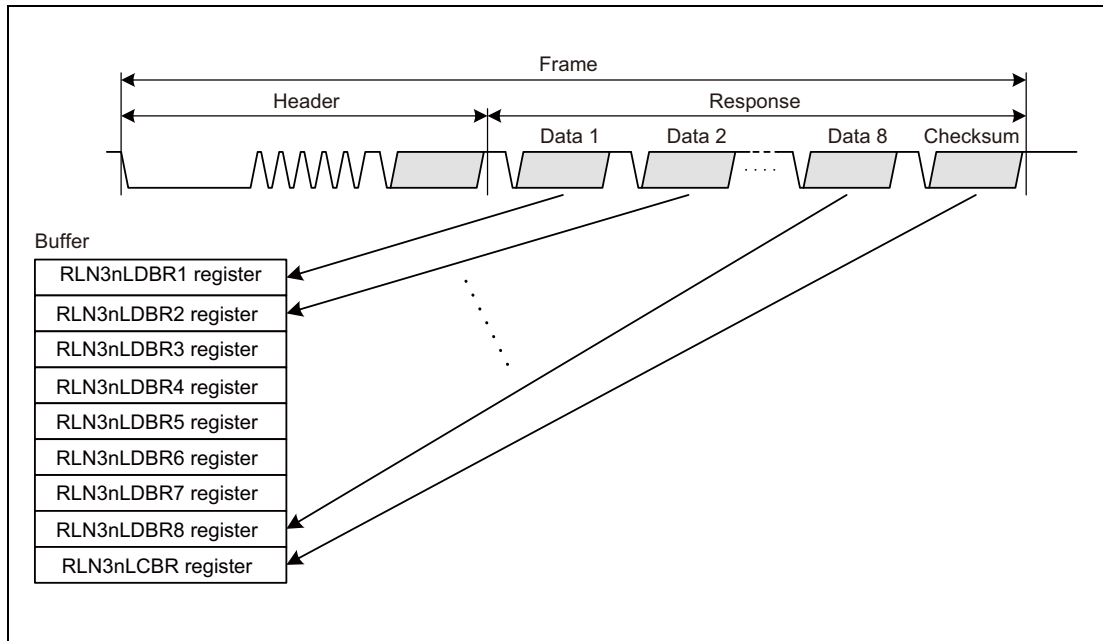


Figure 16.10 LIN Reception Processing and Required Buffer

[Reception of Data 1]

When the reception of the first byte of data is finished, the D1RC flag in the RLN3nLST register is set to 1 (successful data 1 reception).

16.7.3.3 Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses of 10 bytes or more can also be transmitted and received.

In such a case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is longer than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be transmitted or received is not the last data group) in the first data group (variable from 0 to 8 bytes) before transmitting or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the last data group. If it is the last data group, the LSS bit in the RLN3nLDFC register should be set to 0 (indicating that the next data group to be transmitted or received is the last data group), and a checksum should be appended to the last data group.

By changing the RFDL bit in RLN3nLDFC register settings when the RTS bit in RLN3nLTRC register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDFC register to 1 (frame separate mode).

16.7.4 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

16.7.4.1 Wake-up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN3nLDFC register to 1 (LIN master mode: response transmission) and the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low width of the wake-up signal should be set using the WUTL[3:0] bits in the RLN3nLWUP register.

However, if the LWBR0 bit of the RLN3nLWBR register in LIN master mode is 1 (LIN2.x), the LIN system clock (fLIN) becomes low level width at fa regardless of the setting of the LCKS bit of the RLN3nLMD register. By setting the WUTL[3:0] bits of the RLN3nLWUP register to 0100_B (5 Tbits), 260 μs low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN3nLMD register.

If a wake-up low-level is output without any bit error, the FTC flag in the RLN3nLST register is set to 1 (successful frame or wake-up transmission); when the FTCIE bit in the RLN3nLIE register is 1 (successful frame/wakeup transmission interrupt enabled), an interrupt request for RLIN3n transmission is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is canceled and the BER flag in the RLN3nLEST register is set to 1 (bit error detection).

When RLN3nLEDE.PBERE is set in LIN master mode, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

Figure 16.11 shows the wake-up transmission timing.

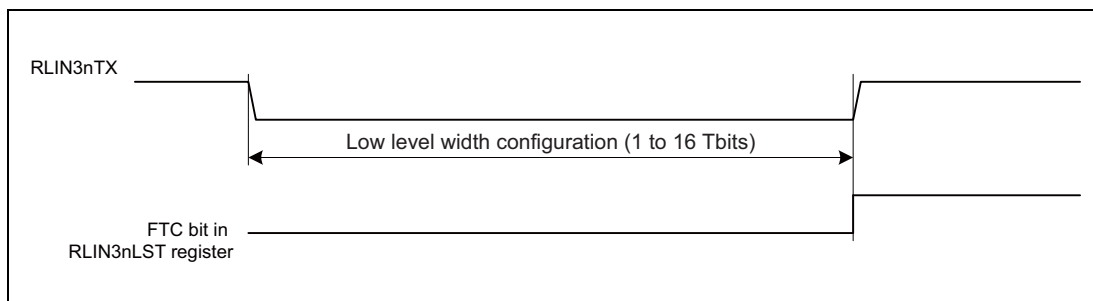


Figure 16.11 Wake-up Transmission Timing

16.7.4.2 Wake-up Reception

To detect a wake-up signal, use the input signal low-level width count function. The input signal low level width count function measures the low level width of the input signal to the RLIN3nRX pin, using the same sampling point as data reception. This allows the 2.5-Tbit or longer low-level width of the input signal of fLIN to be measured.

In LIN master mode, by setting the LWBR0 bit in the RLN3nLWBR register, operation can be executed without changing the baud rate generator setting when switching between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 sets the LIN system clock (fLIN) to fa regardless of the setting of the LCKS bit in the RLN3nLMD register. (The LCKS bit is not changed). By setting the baud rate to 19200bps while fa is selected, an input signal with a low-level width of 130 μ s or longer to be measured regardless of the setting of the LCKS bit in the RLN3nLMD register.

When using the wake-up reception function, in LIN wake-up mode set the RFT bit in the RLN3nLDFC register to 0 (LIN master mode: response reception), or the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register is set to 1 (successful frame/wake-up reception). If the FRCIE bit in the RLN3nLIE register is 1 (successful frame or wake-up reception interrupt enabled), an interrupt request for successful RLIN3n reception is generated.

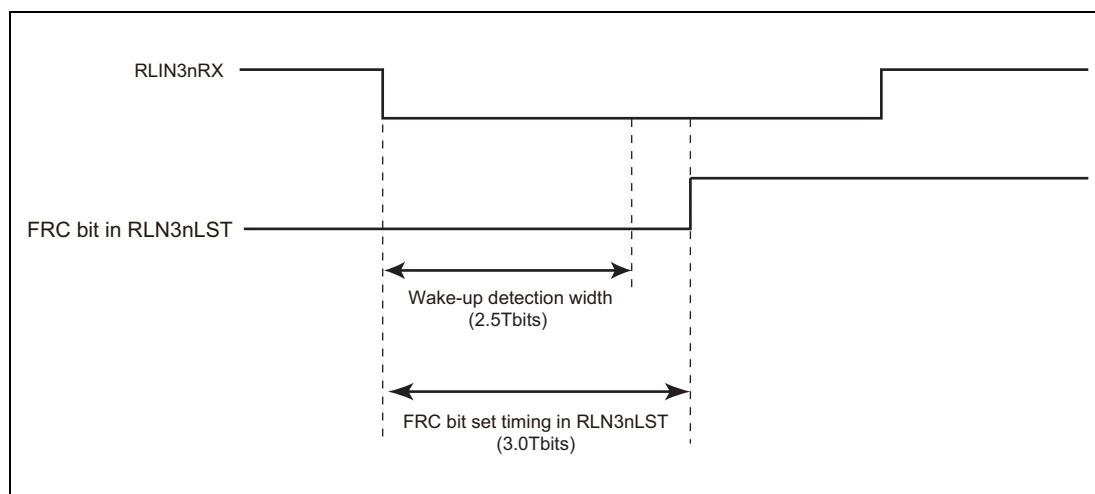


Figure 16.12 Input Signal Low level Count Function

16.7.4.3 Wakeup Collision

If the master node and the slave node transmit wakeup signals simultaneously, a collision will occur on the LIN bus, though a collision of wakeup signals is not detected by the LIN/UART interface.

16.7.5 Status

In LIN mode operation, the LIN/UART interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, and successful header transmission/reception, can generate interrupt requests.

Table 16.63 shows the types of statuses available in LIN master mode.

Table 16.63 Types of Statuses in LIN Master Mode

| Status | Status Set Condition | Status Clear Condition | Operation Mode Capable of Status Detection | Corresponding Bit | Interrupt |
|--------------------------------|--|---|--|--------------------------------|-----------|
| Reset | After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode. | After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode. | All modes | OMM0 bit in RLN3nLMST register | — |
| Operation mode | After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode. | After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode. | <ul style="list-style-type: none"> LIN operation mode LIN wake-up mode | OMM1 bit in RLN3nLMST register | — |
| Frame/wake-up transmission end | When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully. | <ul style="list-style-type: none"> When another communication is started (When the FTS bit in the RLN3nLTRC register is set) When cleared by software After transition to LIN reset mode | <ul style="list-style-type: none"> LIN operation mode LIN wake-up mode | FTC flag in RLN3nLST register | √ |
| Frame/wake-up reception end | When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully. | <ul style="list-style-type: none"> When another communication is started (When the FTS bit in the RLN3nLTRC register is set) When cleared by software After transition to LIN reset mode | <ul style="list-style-type: none"> LIN operation mode LIN wake-up mode | FRC flag in RLN3nLST register | √ |
| Error detection | If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected). | <ul style="list-style-type: none"> When another communication is started (When the FTS bit in the RLN3nLTRC register is set) When cleared by software*1 After transition to LIN reset mode | <ul style="list-style-type: none"> LIN operation mode LIN wake-up mode | ERR flag in RLN3nLST register | √ |
| Data 1 reception end | The RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received successfully.*2 | <ul style="list-style-type: none"> When another communication is started (When the FTS bit in the RLN3nLTRC register is set) When cleared by software After transition to LIN reset mode | LIN operation mode | D1RC flag in RLN3nLST register | — |
| Header transmission end | When a header field is transmitted successfully. | <ul style="list-style-type: none"> When another communication is started (When the FTS bit in the RLN3nLTRC register is set) When cleared by software After transition to LIN reset mode | LIN operation mode | HTRC flag in RLN3nLST register | √ |

Note 1. In LIN operation mode and LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

16.7.6 Error Status

16.7.6.1 LIN Master Mode

(1) Types of Error Statuses

The LIN/UART interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be verified by checking of the corresponding bits in the RLIN3nLEST register.

All error statuses are interrupt factors.

Table 16.64 shows the types of error statuses.

Table 16.64 Types of Error Statuses in LIN Master Mode

| Status | Error Detection Condition | Operation Mode Capable of Error Detection | Communication | Enable/Disable Detection | Corresponding Bit |
|----------------------------|--|--|---------------|--------------------------|----------------------------------|
| Bit error | The transmitted data and the data on the LIN bus monitored by the receive pin do not match ^{*1*2} | <ul style="list-style-type: none"> LIN operation mode LIN wake-up mode | Cancel | Enabled | BER flag in RLIN3nLEST register |
| Physical bus error | <ul style="list-style-type: none"> LIN bus detected a high level when sending a break LIN bus detected a low level when sending a break delimiter LIN bus detected a high level when sending a wake-up | <ul style="list-style-type: none"> LIN operation mode LIN wake-up mode | Cancel | Enabled | PBER flag in RLIN3nLEST register |
| Timeout error | A frame or response transmission/reception is not completed within a given time ^{*3} | LIN operation mode | Cancel | Enabled | FTER flag in RLIN3nLEST register |
| Framing error | In response field reception, a stop bit of each data byte is low level | LIN operation mode | Cancel | Enabled | FER flag in RLIN3nLEST register |
| Checksum error | In response field reception, the result of checksum indicate an error | LIN operation mode | — | Disabled | CSER flag in RLIN3nLEST register |
| Response preparation error | <p>One of the following conditions occurs in frame separate mode during a multi-byte response reception:</p> <ul style="list-style-type: none"> After header transmission is complete, the first byte of receive data is received before a response transmission/reception request is set. After the previous data group reception is complete, the first byte of receive data is received before a transmission/reception request for the next data group is set. | LIN operation mode | Cancel | Disabled | RPER flag in RLIN3nLEST register |

Note 1. If a bit error is detected, the process is canceled after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is canceled immediately after that area. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the bit which had the error is sent.

Note 2. In a multi-byte response transmission, bit errors are detected also between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLIN3nLDFC register) and the checksum selection (the CSM bit in the RLIN3nLDFC register), and this can be calculated according to the following formula:

When the FSM bit in the RLIN3nLDFC register is 1 (frame separate mode), the time-out interval until setting of the RTS bit in the RLIN3nLTRC register is the interval for receiving eight bytes of data. Once the RTS bit is set, the time-out time is reset to the time corresponding to the response field data length (determined by the RFDL[3:0] bits of the RLIN3nLDFC register).

[Frame timeout]

When classic is selected (when the CSM bit in RLIN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]

When enhanced is selected (when the CSM bit in RLIN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced is selected.

[Response timeout]

$$\text{Timeout time} = (\text{number of data bytes} + 1) \times 14 \text{ [Tbit]}$$

When an error is detected, the timeout error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLN3nLTRC register is set), by software, or at a transition to LIN reset mode.

(2) Target Time Area for LIN Error Detection

Figure 16.13 shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

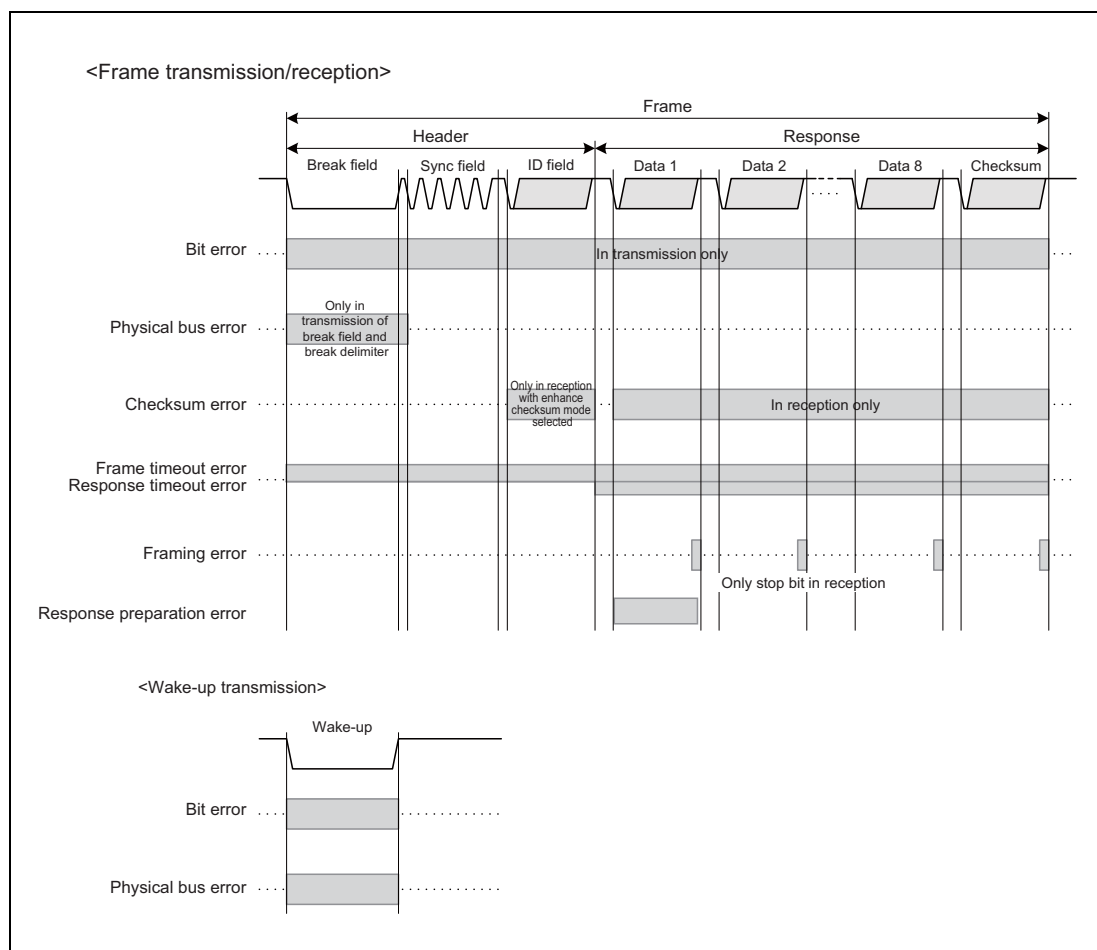


Figure 16.13 Target Time Area for LIN Error Detection (LIN Master Mode)

16.8 UART Mode

In LIN reset mode, setting the LMD bits in the RLIN3nLMD register to 01_B (UART mode) and the OM0 bit in the RLIN3nLCUC register to 1 changes the mode to UART mode, turning the OMM0 bit in the RLIN3nLMST register to 1.

16.8.1 Transmission

Figure 16.14 shows LIN/UART interface (in UART mode) transmission operations; **Table 16.65** shows LIN/UART interface (in UART mode) transmission processing.

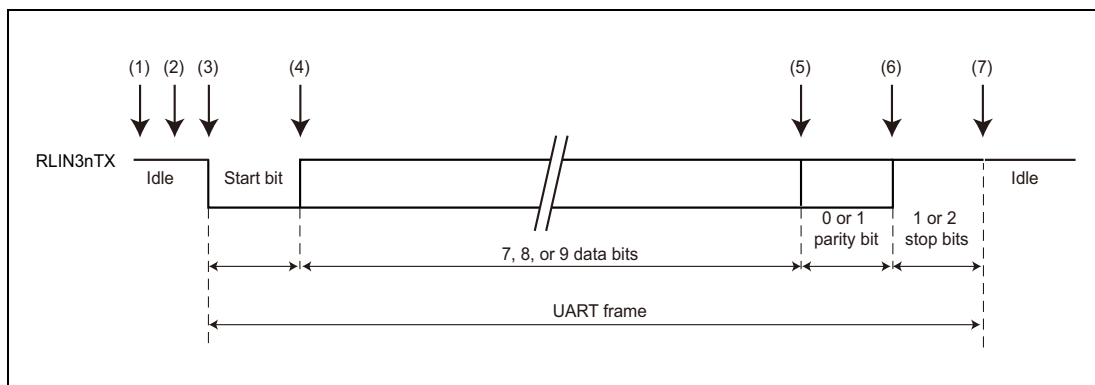


Figure 16.14 LIN/UART Interface (in UART Mode) Transmission Operation

Table 16.65 LIN/UART Interface (UART Mode) Transmission Processing (1/2)

| Software Processing | LIN/UART Interface Processing |
|--|---|
| (1) <ul style="list-style-type: none"> • Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Sets data format. • Sets an interrupt generation timing. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1. | <ul style="list-style-type: none"> • Waits for a transmission trigger (RLIN3nLUTDR register) by software. |
| (2) <ul style="list-style-type: none"> • Sets the transmit data to the UART transmit data register (RLIN3nLUTDR) or UART wait transmit data register (RLIN3nLUWTD). | <ul style="list-style-type: none"> • Sets the transmit status flag. |
| (3) <ul style="list-style-type: none"> • Waits an interrupt request. <p>[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]</p> <ul style="list-style-type: none"> • When transmitting data continuously, sets another piece of transmission data in the UART transmit data register (RLIN3nLUTDR register), waits for the generation of an interrupt request. | <ul style="list-style-type: none"> • Transmits a start bit (for switching between transmission and reception in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see Section 16.8.1.4, Transmission Start Wait Function.) <p>[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]</p> <ul style="list-style-type: none"> • Outputs a transmission interrupt. |
| (4) | Transmits the data set in the UART (for wait) transmit data register. |
| (5) | Transmits a parity bit when parity is used. |
| (6) | Transmits 1 or 2 stop bits. |

Table 16.65 LIN/UART Interface (UART Mode) Transmission Processing (2/2)

| Software Processing | LIN/UART Interface Processing |
|---|--|
| <p>(7) [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]</p> <ul style="list-style-type: none"> • If another item of transmission data is set, goes to step (3). <p>[When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)]</p> <ul style="list-style-type: none"> • When transmitting data continuously, goes to step (2). | <p>[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]</p> <ul style="list-style-type: none"> • If another piece of transmission data is set, goes to step (3). • If another piece of transmission data is not set, clears the transmit status flag. <p>[When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)]</p> <ul style="list-style-type: none"> • Generates RLIN3n transmission interrupt request. • Clears the transmission status flag. |

16.8.1.1 Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLN3nLUTDR register. **Figure 16.15** shows an operation example where the transmission interrupt generation timing is the start of transmission and an operation example where the transmission interrupt generation timing is the end of transmission.

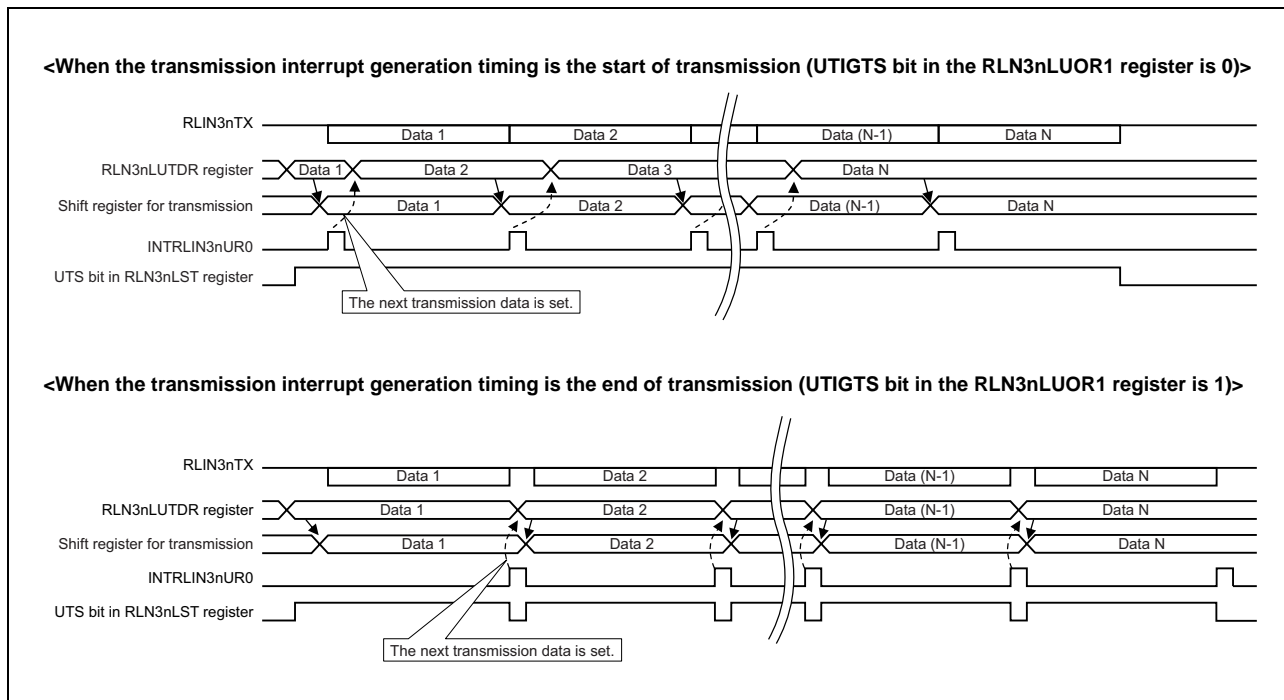


Figure 16.15 Operation Example of LIN/UART Interface (UART Mode) Continuous Transmission

An interrupt can be generated at the end of a transmission by changing the UTIGTS bit in the RLN3nLUOR1 register from 0 to 1 after the start of transmission of final data, provided that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data needs to be known.

16.8.1.2 UART Buffer Transmission

The LIN/UART interface (in UART mode) has a maximum of nine bytes of UART buffers, and thus it is capable of performing continuous transmissions through the use of UART buffers.

Figure 16.16 shows the UART buffer transmission operation of the LIN/UART interface (in UART mode). **Table 16.66** shows the UART buffer transmission processing.

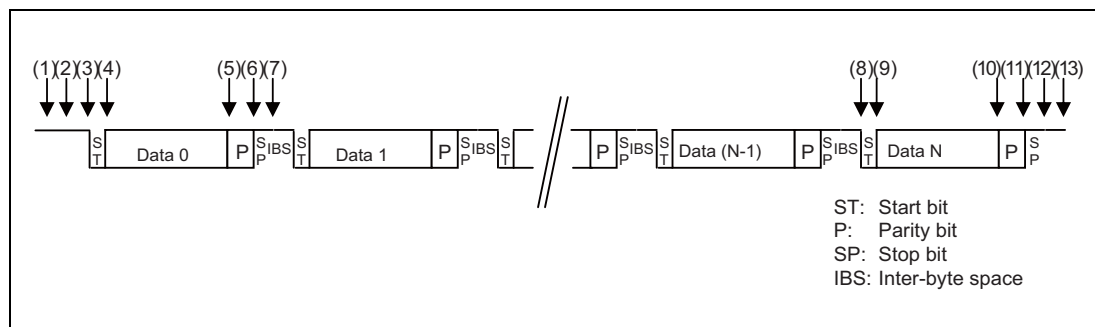


Figure 16.16 UART Buffer Transmission of LIN/UART Interface (in UART Mode)

Table 16.66 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (1/2)

| Software Processing | LIN/UART Interface Processing |
|--|---|
| (1) <ul style="list-style-type: none"> • Sets a baud rate • Sets noise filter ON/OFF • Enables error detection • Sets data format • Sets an interrupt generation timing to the end of transmission. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1 | <ul style="list-style-type: none"> • Waits for a transmission trigger (RTS bit) by software |
| (2) <ul style="list-style-type: none"> • Sets the UART buffer data length and whether the system must wait for the start of transmission. • Sets the transmission data in the UART data 0 buffer register (RLN3nLUDB0) and the LIN data buffer b register (RLN3nLDBRb). (b =1 to 8) • Sets the UART buffer transmission start bit (RTS). | <ul style="list-style-type: none"> • Sets the transmit status flag. |
| (3) Waits for an interrupt request. | Transmits a start bit. (When switching from reception to transmission during half-duplex communication, transmits the start bit upon completion of the stop bit for reception. For details about this function, see Section 16.8.1.4, Transmission Start Wait Function.) |
| (4) | Transmits the data set in the UART data buffer 0 register (RLN3nLUDB0) and the LIN/UART data buffer b register (RLN3nLDBRb). |
| (5) | Transmits a parity bit when parity is used. |
| (6) | Transmits 1 or 2 stop bits (When the number of data set in UART buffer data length select bits is 1, proceeds to (12).) |
| (7) | Transmits an inter-byte space (idle). Repeats steps (3) to (7) until number of data set in the UART buffer data length select bits -1 is reached. |

Table 16.66 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (2/2)

| Software Processing | LIN/UART Interface Processing |
|---------------------|---|
| (8) | Transmits a start bit. |
| (9) | Transmits the data set in the LIN/UART data buffer b register (RLN3nLDBRb). |
| (10) | Transmits a parity bit when parity is used. |
| (11) | Transmits 1 or 2 stop bits. |
| (12) | <ul style="list-style-type: none"> • Sets the successful buffer transmission flag. • Clears the UART buffer transmit start bit (RTS). • A transmission interrupt request signal. • Clears the transmission status flag. |
| (13) | <ul style="list-style-type: none"> • Checks the RLN3nLST register, and clears flags • In the case of continuous data transmission, goes to step (2). |

(1) UART Buffer Transmission

For a 9-byte transmission, the contents stored in the RLN3nLUDB0 and RLN3nLDBR1 to RLN3nLDBR8 registers are transmitted to data areas 0 to 8. The RLN3nLUDB0 register is used only if 9-byte transmission is set. In other cases, the RLN3nLDBR1 to RLN3nLDBR8 registers are selected depending upon the length of data involved. For a 4-byte transmission, the contents stored in the RLN3nLDBR1 to RLN3nLDBR4 registers are transmitted to data areas 1 to 4, but the contents of the RLN3nLDBR5 to RLN3nLDBR8 registers are not transmitted. An RLIN3n transmission interrupt is generated after the number of data specified in the MDL [3:0] bits of the RLN3nLDFC register is transmitted. The spaces between transmission data items can be set in the IBS bit in the RLN3nLSC register.

Figure 16.17 shows a 9-byte UART buffer and the transmission processing.

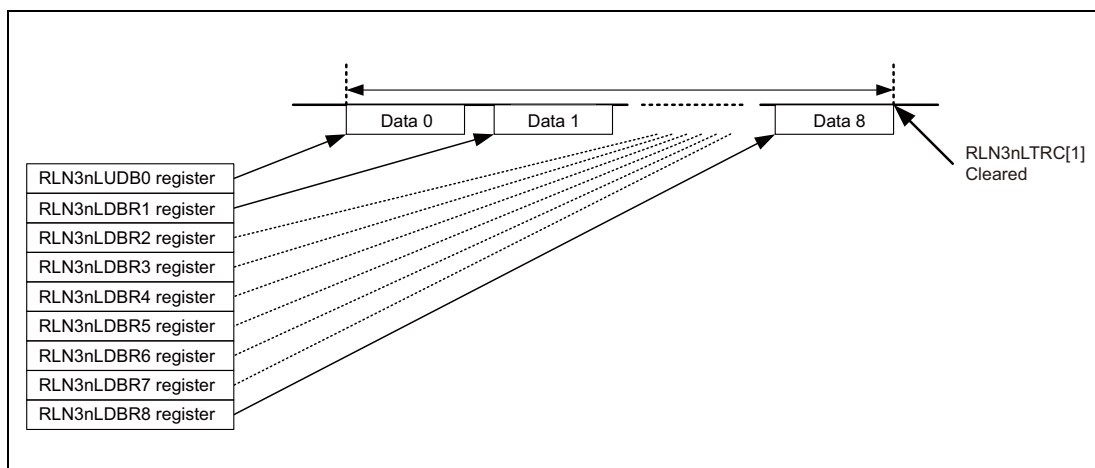


Figure 16.17 UART Buffer and Transmission Processing (for 9-Byte Transmission)

16.8.1.3 Data Transmission

One bit of data is transmitted per Tbit.

In half-duplex communication, if the BERE bit in the RLN3nLEDE register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 16.8.5, Error Status**). The timing at which the input pin is sampled during data transmission can vary depending upon the settings of the LPRS[2:0] and NSPB[3:0] bits in the RLN3nLWBR register.

The bit error detection timing in UART mode is shown in **Table 16.67**.

Table 16.67 Error Detection Timing in UART Mode

| Sampling Count Per Bit | Bit Error Detection Timing |
|--|-------------------------------------|
| 6 samples | 3rd clock cycle + 1 prescaler clock |
| 7 samples | 4th clock cycle + 1 prescaler clock |
| 8 samples | 4th clock cycle + 1 prescaler clock |
| 9 samples | 5th clock cycle + 1 prescaler clock |
| 10 samples | 5th clock cycle + 1 prescaler clock |
| 11 samples | 6th clock cycle + 1 prescaler clock |
| 12 samples | 6th clock cycle + 1 prescaler clock |
| 13 samples | 7th clock cycle + 1 prescaler clock |
| 14 samples <td 7th clock cycle + 1 prescaler clock | |
| 15 samples | 8th clock cycle + 1 prescaler clock |
| 16 samples | 8th clock cycle + 1 prescaler clock |

Example of Data Transmission Timing (when 1 Tbit = 16 Sampling) is shown in **Figure 16.18**

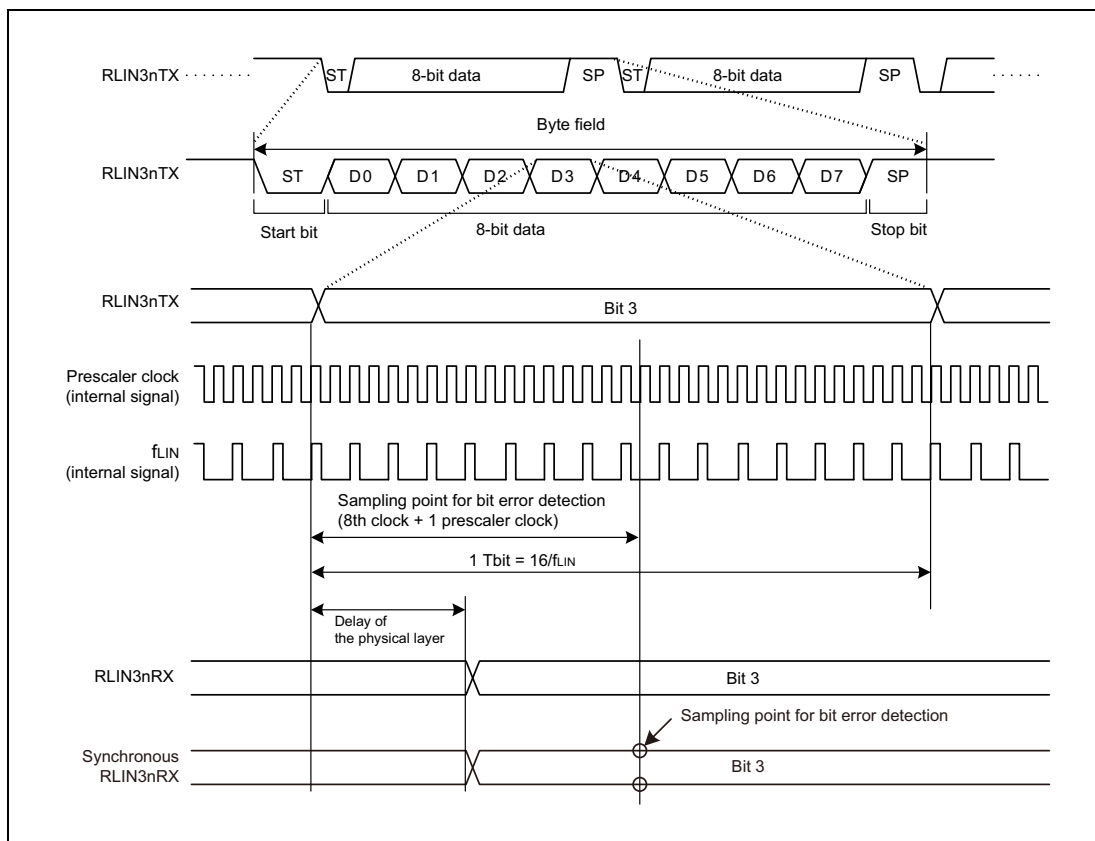


Figure 16.18 Example of Data Transmission Timing (When 1 Tbit = 16 Sampling)

16.8.1.4 Transmission Start Wait Function

For performing half-duplex communication, the LIN/UART interface (in UART mode) has the function of securing the reception stop bit length when switching from reception to transmission.

If it is desired to delay the start of transmission until the stop bits for the reception are completed, set data in the RLIN3nLUWTDR register, which is used only for the wait function, instead of setting transmission data in the RLIN3nLUTDR register as a start-of-transmission request. When transmitting from the UART buffer, set 1 (UART buffer transmission started) in the RTS bit in the RLIN3nLTRC register with 1 set in the UTSW bit in the RLIN3nLDFC register.

In such a case, the LIN/UART interface delays the start of transmission until the stop bits of reception data are completed.

Note that even if the UART stop bit length selection bit (USBLS) in RLIN3nBLFC register is 1 (stop bits = 2 bits), there is only a 1-bit delay.

Figure 16.19 shows the operation of transmission wait function.

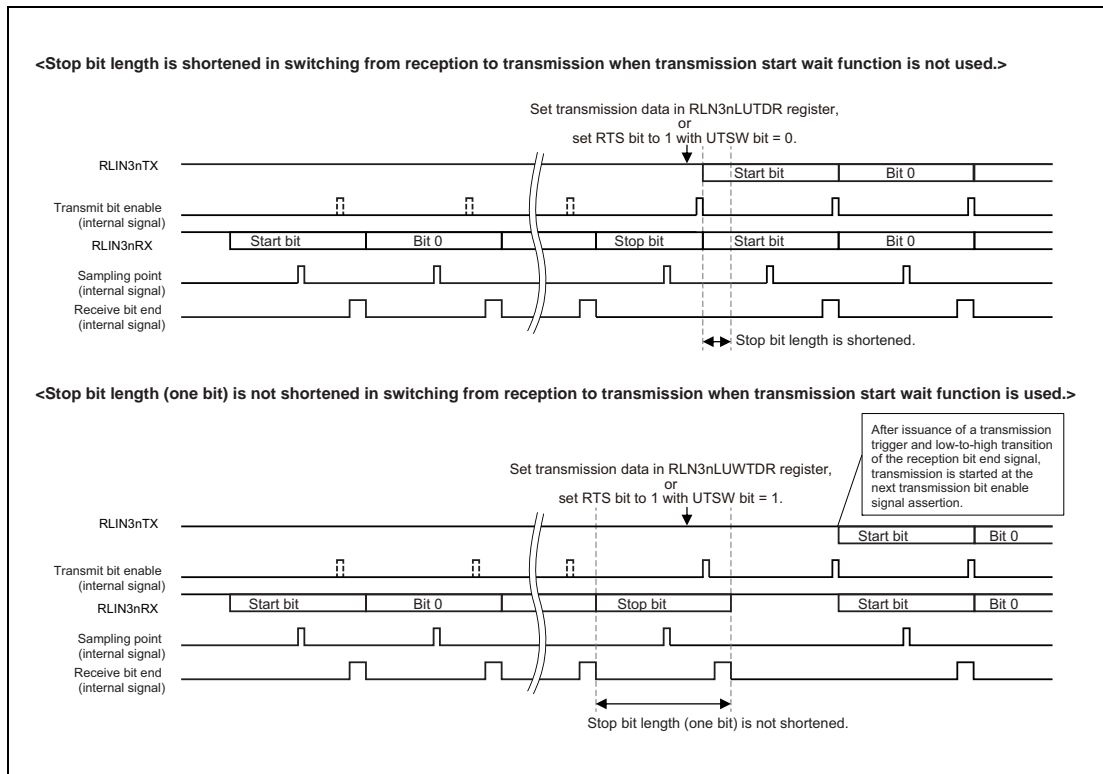


Figure 16.19 Example of Data Transmission Timing (When 1 Tbit = 16 samplings)

16.8.2 Reception

Figure 16.20 shows the LIN/UART interface (in UART mode) reception operation. Table 16.68 shows the LIN/UART interface (in UART mode) reception processing.

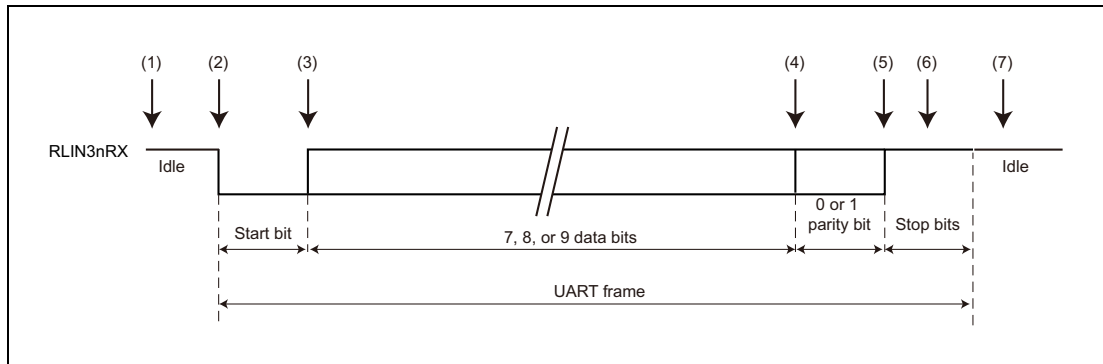


Figure 16.20 LIN/UART Interface (in UART Mode) Reception Operation

Table 16.68 LIN/UART Interface (in UART Mode) Reception Processing

| Software Processing | LIN/UART Interface Processing |
|---|--|
| (1) <ul style="list-style-type: none"> • Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Sets data format. • Clears the LIN/UART interface from LIN reset mode. • Sets the receive enable bit (UROE bit) to 1. | <ul style="list-style-type: none"> • Waits for the reception to be enabled by software. • Waits for detection of a start bit. |
| (2) Waits for an interrupt request. | <ul style="list-style-type: none"> • Waits for a falling edge from the reception pin, and detects a start bit. • Sets the reception status flag. |
| (3) | Receives data. |
| (4) | Receives a parity bit when parity is used. |
| (5) | Receives only 1 stop bit. |
| (6) | <ul style="list-style-type: none"> • Generates a RLIN3n reception complete interrupt request. • Clears the reception status flag. |
| (7) Checks the RLN3nLST register, and clears flags | Waits for a falling edge from the reception pin. |

16.8.2.1 Data Reception

Data reception is performed by using the synchronized RLIN3nRX (an internal signal) that is the input from the RLIN3nRX pin synchronized with the prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, resampling is performed 0.5 Tbits later when the number of sampling per 1 Tbit is even and $\{(the\ number\ of\ sampling + 1) / 2\}$ / (the number of sampling) Tbits later when the number is odd. If the synchronized RLIN3nRX signal is low level, the bit is recognized as a start bit. The bit is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted or if a high level is detected during the resampling.

After the start bit is detected, 1 bit is sampled per Tbit.

However, when the BERE bit in the RLN3nLEDE register is 1, the sampling point is the same as the bit error detection timing.

The LIN/UART interface has a noise filter function for received data. If the LRDNFS bit in the RLN3nLMD register is 0, the noise filter is used. For a sampling value, the value determined by a 3-sampling majority rule by the prescaler clock is used. If the LRDNFS bit in the RLN3nLMD register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 16.21 shows an example of data reception timing.

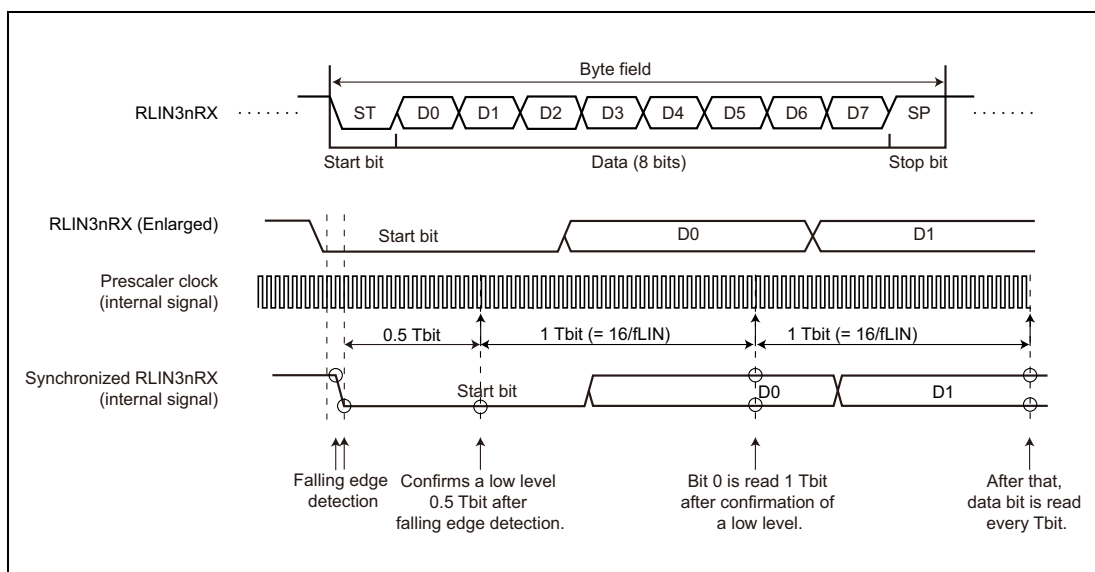


Figure 16.21 Example of Data Reception Timing (When Sampling Count is 16 in 1 Tbit)

16.8.3 Expansion Bits

The LIN/UART interface (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the RLIN3nLUOR1 register to 1.

16.8.3.1 Expansion Bit Transmission

The LIN/UART interface (in UART mode) can transmit 9-bit long data when the expansion bit enable bit (UEBE) in the UART option register 1 (RLIN3nLUOR1) is 1 and by writing the 9-bit data to either the UART transmission data register (RLIN3nLUTDR) or the UART wait transmission data register (RLIN3nLUWTD).

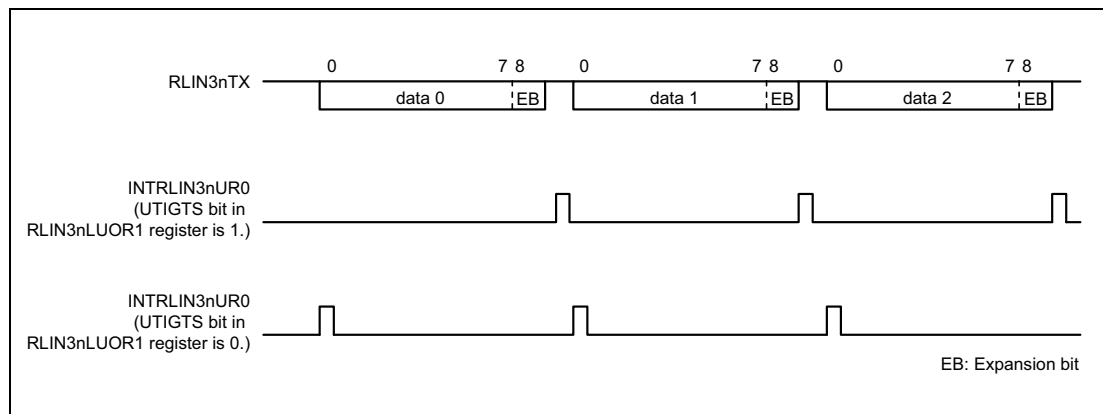


Figure 16.22 Transmission Example When Expansion Bit is Enabled (LSB First)

16.8.3.2 Expansion Bit Reception

With the LIN/UART interface (in UART mode), 9-bit data can always be received without requiring a comparison of expansion bits, provided that the expansion bit enable bit (UEBE) in the UART option register 1 (RLIN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit data comparison enable bit (UEBDCE) is 0. Irrespective of the particular setting of the expansion bit detection level select bit (UEBDL) in the UART option register 1 (RLIN3nLUOR1), a successful RLIN3n reception interrupt is generated when 9-bit data is received.

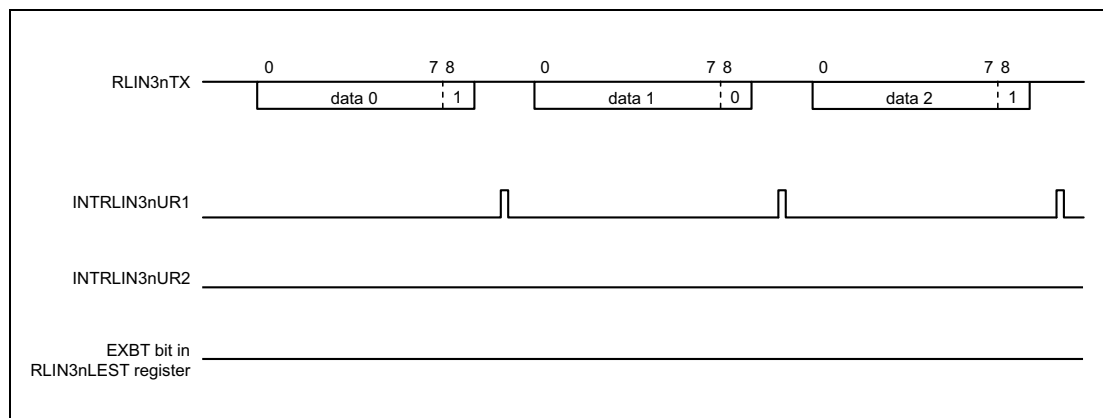


Figure 16.23 Expansion Bit Reception Example (LSB First)

16.8.3.3 Expansion Bit Reception (with Expansion Bit Comparison)

The LIN/UART interface (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 0.

If the level that was set in the expansion bit detection level select bit (UEBDL) is detected, an RLIN3n status interrupt request is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN error status register (RLN3nLEST) is set. If the reversed value of an expansion bit detection level is detected, RLIN3n reception complete interrupt request is generated. In either case, the received data is stored in the UART reception data register (RLN3nLURDR), unless there was an overrun error.

Figure 16.24 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

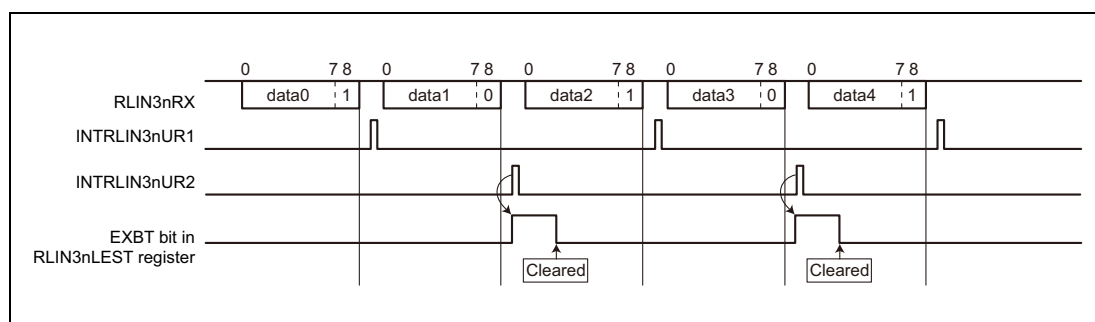


Figure 16.24 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)

NOTE

- If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. In this case, a successful RLIN3n reception interrupt is not generated.
- If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs, the expansion bit detection flag (EXBT) is also set.

16.8.3.4 Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 1, and if the level that was set by the expansion bit detection level select bit (UEBDL) is detected, the LIN/UART interface (in UART mode) compares the 8 bits, exclusive of the expansion bit in the received data, with the a pre-set RLN3nLIDB register value.

If the result of the comparison is a match, the LIN/UART interface performs the following operations:

- Generates an RLIN3n status interrupt
- Sets an expansion bit detection flag (EXBT)
- Sets an ID match flag (IDMT)
- Stores the received data in the UART reception data register (RLN3nLURDR)

Even when the result of the comparison is a match, successful RLIN3n reception interrupt is not generated.

If the result of the comparison is not a match, no successful RLIN3n reception interrupt or RLIN3n status interrupt is generated, and the EXBT and IDMT flags are not set to 1. The received data is not stored in the UART reception data register (RLN3nLURDR).

When changing the UEBDCE bit to 0, make the change before the reception of another set of data is finished.

Figure 16.25 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

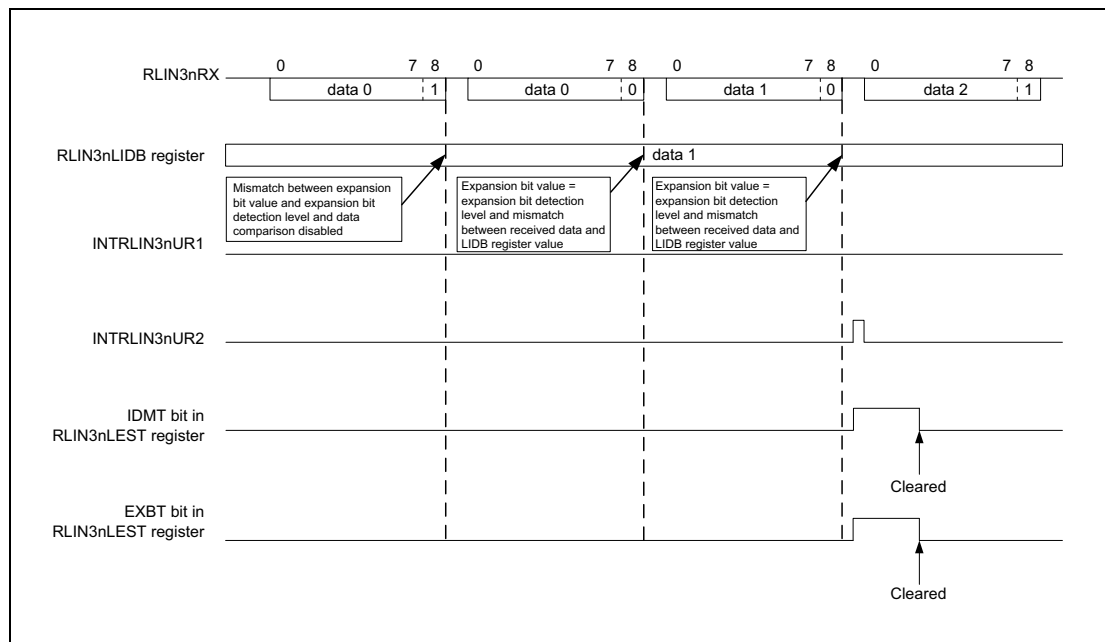


Figure 16.25 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)

NOTE

When a reception error (parity, framing, or overrun) occurs, a status interrupt of RLIN3n is generated and the error flag is updated. When the overrun error occurs and the comparison result matches, the EXBT and IDMT flags are also set.

16.8.4 Status

In UART mode, the LIN/UART interface can detect five types of statuses.

Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.

Table 16.69 shows the types of statuses available in UART mode.

Table 16.69 Types of Statuses in UART Mode

| Status | Status Set Condition | Status Clear Condition | Corresponding Bit | Interrupt |
|-------------------------------------|---|---|--------------------------------|-----------|
| Reset | After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode. | After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode. | OMM0 bit in RLN3nLMST register | — |
| Successful UART buffer transmission | <ul style="list-style-type: none"> When the UTIGTS bit in the RLN3nLUOR1 register is 0 (transmission interrupt request is generated upon start of transmission), the transmission of the last data of the data length set by the MDL bit in the RLN3nLDFC register is started. When the UTIGTS bit in the RLN3nLUOR1 register is 1 (transmission interrupt request is generated upon end of transmission), the transmission of the data length set by the MDL bit in the RLN3nLDFC register is ended. | <ul style="list-style-type: none"> When cleared by software Transition to LIN reset mode | FTC flag in RLN3nLST register | √ |
| Error detection | If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected). | <ul style="list-style-type: none"> When cleared by software*¹ Transition to LIN reset mode | ERR flag in RLN3nLST register | √ |
| Transmission status | <ul style="list-style-type: none"> When data is written to the RLN3nLUTDR or RLN3nLUWTDRC register. When 1 is written to the RTS bit in the RLN3nLTRC register. | <ul style="list-style-type: none"> The transmission of the data set in the RLN3nLUTDR or RLN3nLUWTDRC register is complete, but another transmission data item is not set The transmission of the data in the UART buffer is complete, and the RTS bit in the RLN3nLTRC register is cleared Transition to LIN reset mode | UTS flag in RLN3nLST register | — |
| Reception status | <ul style="list-style-type: none"> When a start bit is detected. | <ul style="list-style-type: none"> When a sampling point for stop bits is detected Transition to LIN reset mode | URS flag in RLN3nLST register | — |

Note 1. Writing a 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register when the LIN reset mode is being canceled sets the ERR flag in the RLN3nLST register to 0.

16.8.5 Error Status

Types of Error Statuses

In UART mode, the LIN/UART interface can detect four types of errors and two types of statuses. The condition of these statuses can be verified by using the corresponding bits in the RLIN3nLEST register.

Table 16.70 shows available status types.

Table 16.70 Types of Statuses in UART Mode

| Status | Error Detection Condition | Communication | Enable/Disable Detection | Corresponding Bit |
|-------------------------|---|--|--------------------------|----------------------------------|
| Bit error | The transmitted data and the data monitored on the receive pin do not match ¹ | Continues until the transmission of the set transmission data is finished. | Enabled | BER flag in RLIN3nLEST register |
| Overrun error | After received data is stored in the RLIN3nLURDR register, another data item is received before the data is read. (In this case, no data is stored in the RLIN3nLURDR register). | — (Reception is finished by the time this error is detected) | Enabled | OER flag in RLIN3nLEST register |
| Framing error | When the first stop bit is low level in the reception processing. | — (Reception is finished by the time this error is detected) | Enabled | FER flag in RLIN3nLEST register |
| Parity error | The received parity value fails to match the parity value calculated from the received data | Continues until the data reception is finished. | Disabled ² | UPER flag in RLIN3nLEST register |
| Expansion bit detection | The value of the received expansion bit matches the value of the UEEDL bit in the RLIN3nLUOR1 register. | — | Enabled | EXBT flag in RLIN3nLEST register |
| ID match detection | The value of the received expansion bit matches the value of the UEEDL bit in the RLIN3nLUOR1 register and the 8-bit receive data excluding the expansion bit matches the value of the RLIN3nLIDB register. | — | Enabled | IDMT flag in RLIN3nLEST register |

Note 1. In the case of transmission from the UART buffer, bit errors are detected even in the space between UART frames (inter-byte space).

Note 2. Setting the UPS[1:0] bits in the RLIN3nLBFC register to 10_B (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

The error status is cleared by software or at a transition to LIN reset mode.

16.9 LIN Self-Test Mode

The LIN/UART interface provides a LIN self-test mode.

When the LIN/UART interface enters the LIN self-test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and RLIN3nTX and RLIN3nRX are connected to the LIN/UART interface internally. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX. The LIN self-test mode can perform tests exclusively in LIN mode.

The self-test can be performed in the following two modes:

- LIN master self-test mode (transmission): Header transmission and response transmission
- LIN master self-test mode (reception): Header transmission and response reception

In LIN self-test mode, operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate setting is the LIN communication clock source/16 [bps]. (The NSPB bits in the RLIN3nLWBR register should be set to 0000_B or 1111_B.)

In addition, in LIN self-test mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- Frame/response timeout error

Do not use these functions.

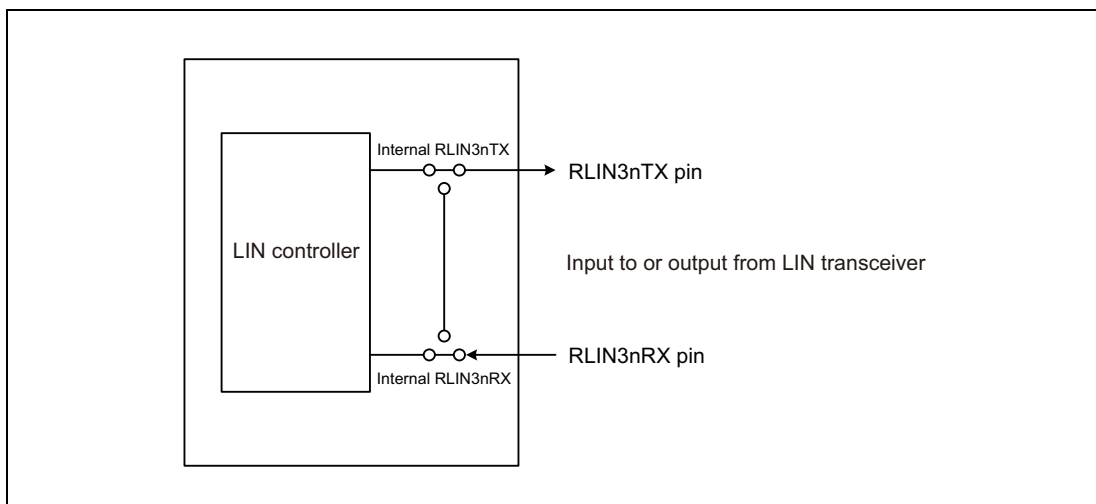


Figure 16.26 Connection in LIN Reset Mode, LIN Mode, and UART Mode

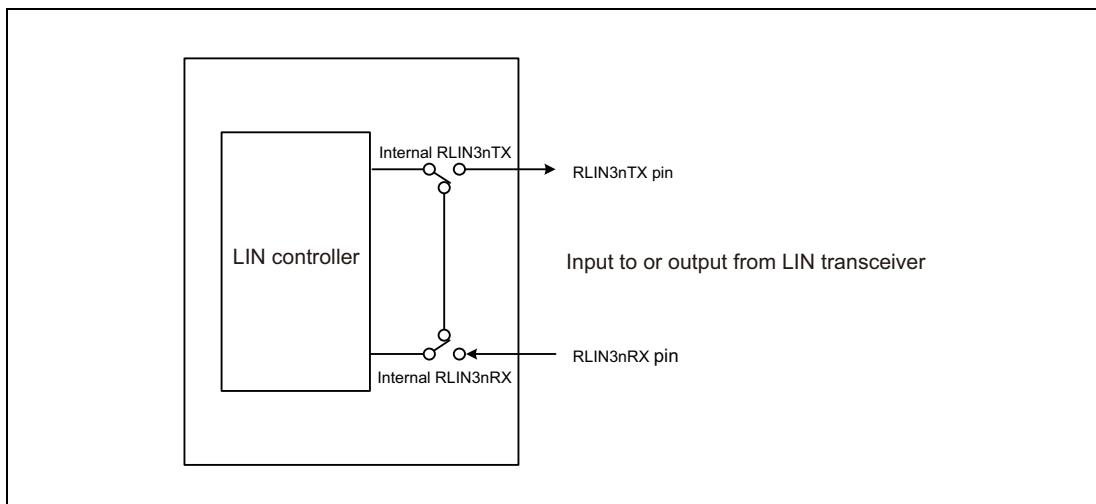


Figure 16.27 Connection in LIN Self-Test Mode

16.9.1 Transitioning to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.

The LSTM bit in the RLN3nLSTC register set to 1 indicates that the mode has transitioned to the LIN self-test mode.

To transition to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be written three times consecutively to the LIN self-test control register, as shown below:

- Transition to LIN reset mode
Set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode).
Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).
- Select a LIN mode
LMD bits in RLN3nLMD = 00_B (LIN master mode)
1st write: RLN3nLSTC register = 1010 0111 (A7_H)
- 2nd write: RLN3nLSTC register = 0101 1000 (58_H)
- 3rd write: RLN3nLSTC register = 0000 0001_B (01_H)
- Verify the transition to LIN self-test mode
Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the first write (A7_H) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the step of first write. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also canceled.

16.9.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
 RLN3nLWBR register = 0000 xxxx_B^{*1}
 RLN3nLBRP0 register = xxxx xxxx_B^{*1}
 RLN3nLBRP1 register = xxxx xxxx_B^{*1}
 RLN3nLMD register = 00xx xx00_B^{*1}
- Set the interrupt enable and error enable related registers.
 RLN3nLIE register = 0000 xxxx_B^{*2}
 RLN3nLEDE register = x000 x0xx
- Set the break field and space related registers.
 RLN3nLBFC register = 00xx xxxx_B
 RLN3nLSC register = 00xx 0xxx_B
- Cancel the LIN reset mode.
 Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the transmit frame related registers.
 RLN3nLDFC register = 00x1 xxxx_B
 RLN3nLIDB register = xxxx xxxx_B
 RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx xxxx_B
- Header transmission → response transmission started
 Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).
 The LIN master self-test mode (transmission) is executed. In this mode, interrupt are generated, and status and error status are also updated. The checksum is automatically calculated by the LIN/UART interface.
 To suspend the transmission during LIN master self-test mode (transmission), write 0 to the OM0 bit of the RLN3nLCUC register (LIN reset mode) and transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register and the LCKS bit in the RLN3nLMD register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupt**.

Note 3. When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).

The time required from the setting of the successful header transmission flag to the setting of the successful frame/wake-up transmission flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) [\text{Tbit}]$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN system clock source (fLIN)} \times 16$$

16.9.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
 - RLN3nLWBR register = 0000 xxxx_B^{*1}
 - RLN3nLBRP0 register = xxxx xxxx_B^{*1}
 - RLN3nLBRP1 register = xxxx xxxx_B^{*1}
 - RLN3nLMD register = 00xx xx00_B^{*1}
- Set the interrupt enable and error enable related registers.
 - RLN3nLIE register = 0000 xxxx_B^{*2}
 - RLN3nLEDE register = x000 x0xx_B
- Set the break field and space related registers.
 - RLN3nLBFC register = 00xx xxxx_B
 - RLN3nLSC register = 00xx 0xxx_B^{*1}
- Cancel the LIN reset mode.
 - Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the reception frame related registers.
 - RLN3nLDFC register = 00x0 xxxx_B
 - RLN3nLIDB register = xxxx xxxx_B
 - RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx xxxx_B
 - RLN3nLCBR register = xxxx xxxx_B

Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.
- Header transmission → response reception started
 - Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).
 - The LIN master self-test mode (reception) is executed. In this mode, interrupt are generated, and status and error status are also updated.
 - To suspend the transmission during LIN master self-test mode (reception), write 0 to the OM0 bit of the RLN3nLCUC register (LIN reset mode) and transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1

register, the LCKS bit in the RLN3nLMD register, and the IBS bit in the RLN3nLSC register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupt**.

Note 3. When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).

The time required from the setting of the successful header transmission flag to the setting of the successful frame/wake-up reception flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) \text{ [Tbit]}$$

$$1 \text{ Tbit} = 1/\text{LIN system clock source (fLIN)} \times 16$$

16.9.4 Terminating LIN Self-Test Mode

To terminate LIN self-test mode, perform the procedure below:

- Write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register.
If the OMM1 and OMM0 bits in the RLN3nLMST register are not 11_B, write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11_B, transition to LIN reset mode.
- Verify the cancelation of LIN self-test mode.
Read the LSTM bit in the RLN3nLSTC register; confirm that it is 0 (not in LIN self-test mode).
- Verify the transition to LIN reset mode.
Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).

16.10 Baud Rate Generator

The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock (f_{LIN}) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock (f_{LIN}) by the number of samplings is the baud rate. The reciprocal of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two types of baud rate generators. The baud rate generator to be used is switched according to the mode.

16.10.1 LIN Master Mode

Figure 16.28 shows a block diagram of baud rate generation in LIN master mode.

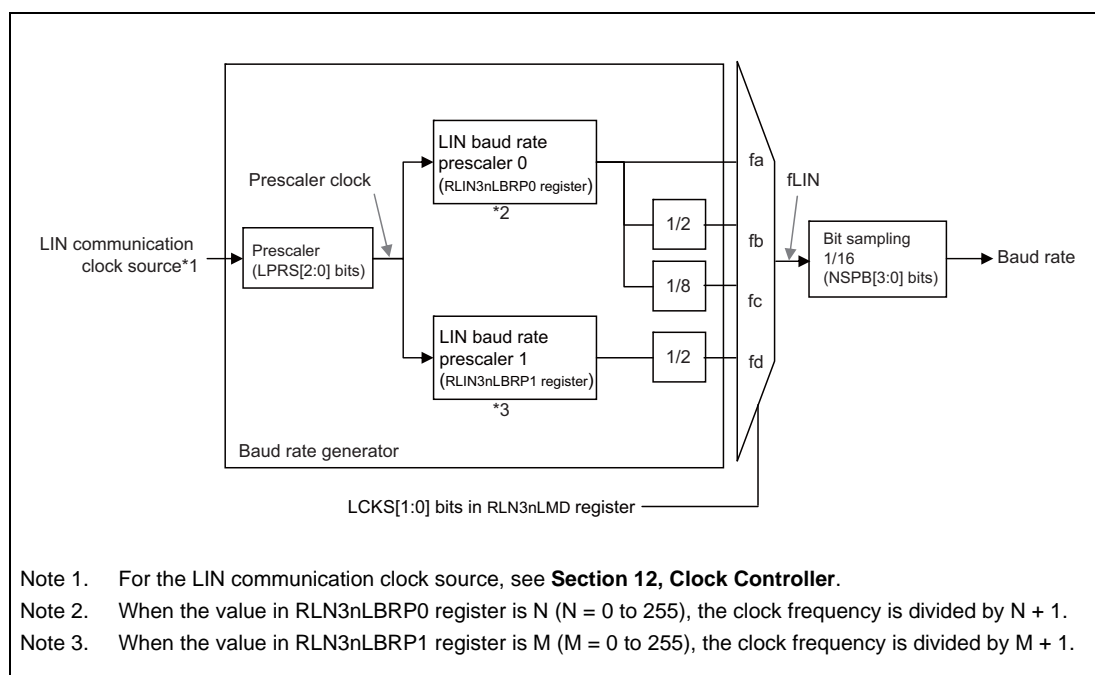


Figure 16.28 Block Diagram of Baud Rate Generation in LIN Master Mode

By setting the RLIN3nLBRP0 register so that f_a is 307200 Hz ($= 19200 \times 16$), the resulting f_{LIN} are $f_a = 19200 \times 16$, $f_b = 9600 \times 16$, and $f_c = 2400 \times 16$. These f_{LIN} by 16 in the bit timing generator, enabling bit rates of 19200 bps, 9600 bps and 2400 bps, to be generated. Also, by setting the RLIN3nLBRP1 register so that f_d is 166672 Hz ($= 10417 \times 16$), the resulting f_{LIN} is $f_d = 10417 \times 16$. This f_{LIN} by 16 in the bit timing generator, enabling 10417 bps to be generated.

Table 16.71 shows examples of baud rate generation for each frequency of the LIN communications clock source (19200, 10417, 9600, and 2400 bps) in LIN master mode and the sizes of errors in the rates.

**Table 16.71 Example of Baud Rate Generation in LIN master mode
(19200 bps, 10417 bps, 9600 bps, and 2400 bps)**

| LIN Communication Clock Source | Prescaler | Baud Rate Generator 0 (N + 1) Dividing | Baud Rate Generator 0 (N + 1) Dividing | System Clock | Baud Rate | Error |
|--------------------------------|-----------|--|--|--------------|-----------|---------|
| 80 MHz | 1/2 | 130 | — | fa | 19230.77 | +0.16% |
| | 1/1 | — | 240 | fd | 10416.67 | -0.003% |
| | 1/2 | 130 | — | fb | 9615.38 | +0.16% |
| | 1/2 | 130 | — | fc | 2403.85 | +0.16% |

NOTE

The number of samples per bit is 16 (when RLN3nLWBR.NSPB[3:0] = 0000_B or 1111_B).

Baud rate is calculated by the following formula:

Baud rate of LIN slave

$$= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS}[2:0] \text{ selection clock}) \\ \div (\text{RLN3nLBRP0} + 1) \div 16 \text{ [bps]} \text{ (When fa is selected for fLIN)}$$

$$= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS}[2:0] \text{ selection clock}) \\ \div (\text{RLN3nLBRP0} + 1) \div 2 \text{ or } 16 \text{ [bps]} \text{ (When fb is selected for fLIN)}$$

$$= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS}[2:0] \text{ selection clock}) \\ \div (\text{RLN3nLBRP0} + 1) \div 8 \div 16 \text{ [bps]} \text{ (When fc is selected for fLLIN)}$$

$$= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS}[2:0] \text{ selection clock}) \\ \div (\text{RLN3nLBRP1} + 1) \div 2 \div 16 \text{ [bps]} \text{ (When fd is selected for fLIN)}$$

16.10.2 UART Mode

Figure 16.29 shows a block diagram of baud rate generation in UART mode.

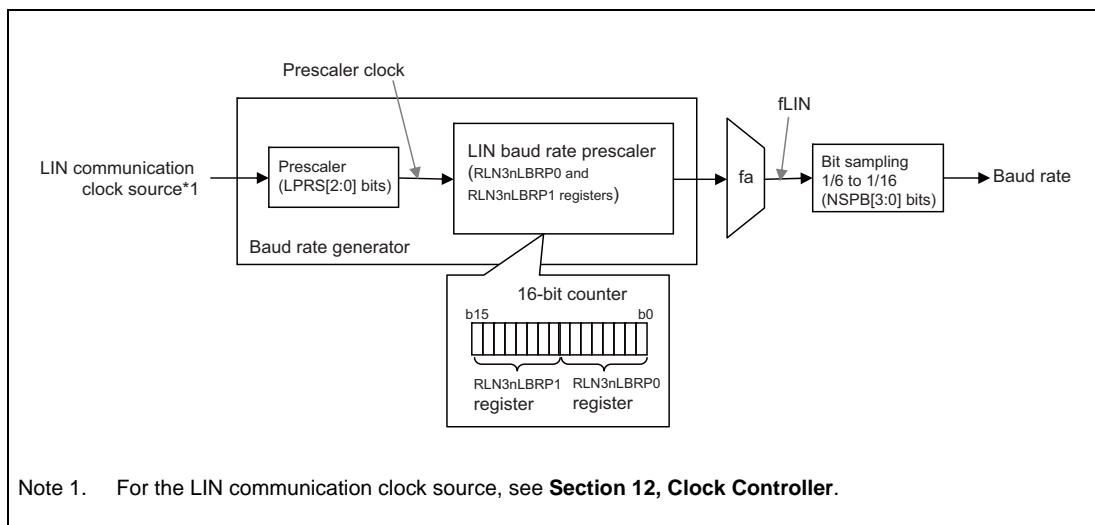


Figure 16.29 Block Diagram of Baud Rate Generation in UART Mode

UART baud rate is calculated with the following formula:

$$\begin{aligned} &\text{UART baud rate} \\ &= \{ \text{LIN communication clock source frequency} \} \times (\text{RLN3nLWBR.LPRS}[2:0] \text{ select clock}) \div \\ &(\text{RLN3nLBRP0} + 1) \div \{ \text{RLN3nLWBR.NSPB}[3:0] \text{ select count} \} [\text{bps}] \end{aligned}$$

Table 16.72 shows examples of baud rate generation for each frequency of the LIN communications clock source (19200, 9600, 2400, and 10417 bps) and the sizes of the errors in the rates.

Table 16.72 Examples of UART Baud Rate Setting (When LIN Communication Clock Source is 80 MHz)

| UART Baud Rate (Target) | Prescaler | Baud Rate Generator 01 (L + 1) Dividing | Baud Rate | Error |
|-------------------------|-----------|---|-----------|--------|
| 1200 bps | 1/2 | 2084 | 1199.616 | -0.03% |
| 2400 bps | 1/2 | 1042 | 2399.232 | -0.03% |
| 4800 bps | 1/2 | 520 | 4807.692 | +0.16% |
| 9600 bps | 1/2 | 260 | 9615.385 | +0.16% |
| 19200 bps | 1/2 | 130 | 19230.77 | +0.16% |
| 31250 bps | 1/2 | 80 | 31250.00 | 0.00% |
| 38400 bps | 1/2 | 66 | 37878.79 | -1.36% |

NOTE

The number of samples per bit in Table 16.72 is 16 (when RLN3nLWBR.NSPB[3:0] = 0000_B or 1111_B).

16.11 Noise Filter

The LIN/UART interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLIN3nLMD register to 0 (use the noise filter), the noise filter is enabled. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-sampling majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 16.30 shows the configuration of the noise filter, **Figure 16.31** shows an example of a noise filter circuit, and **Figure 16.32** shows the determination of the received data when the noise filter is used.

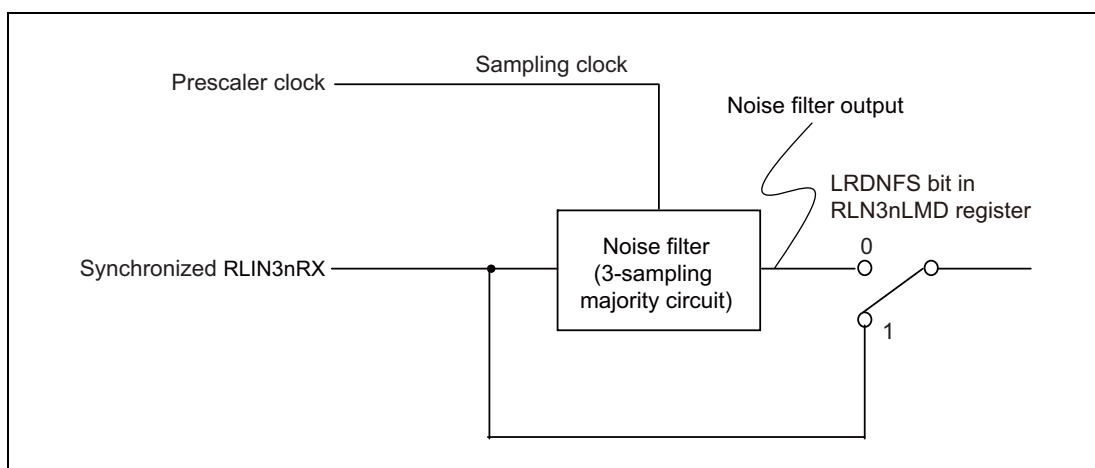


Figure 16.30 Configuration of Noise Filter

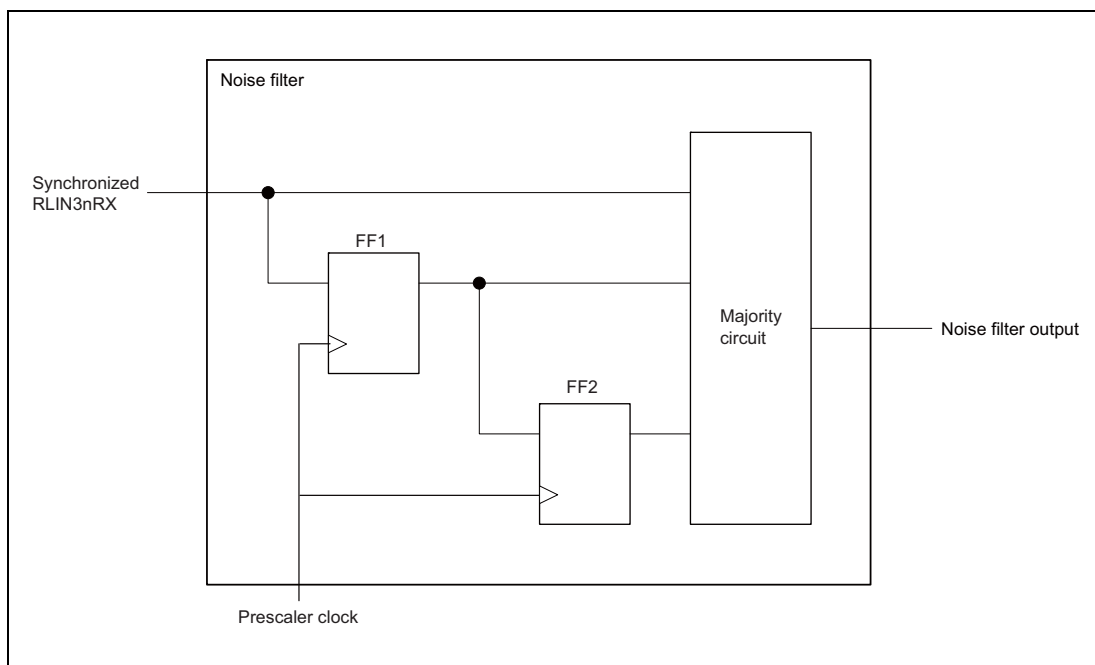


Figure 16.31 Example of Noise Filter Circuit

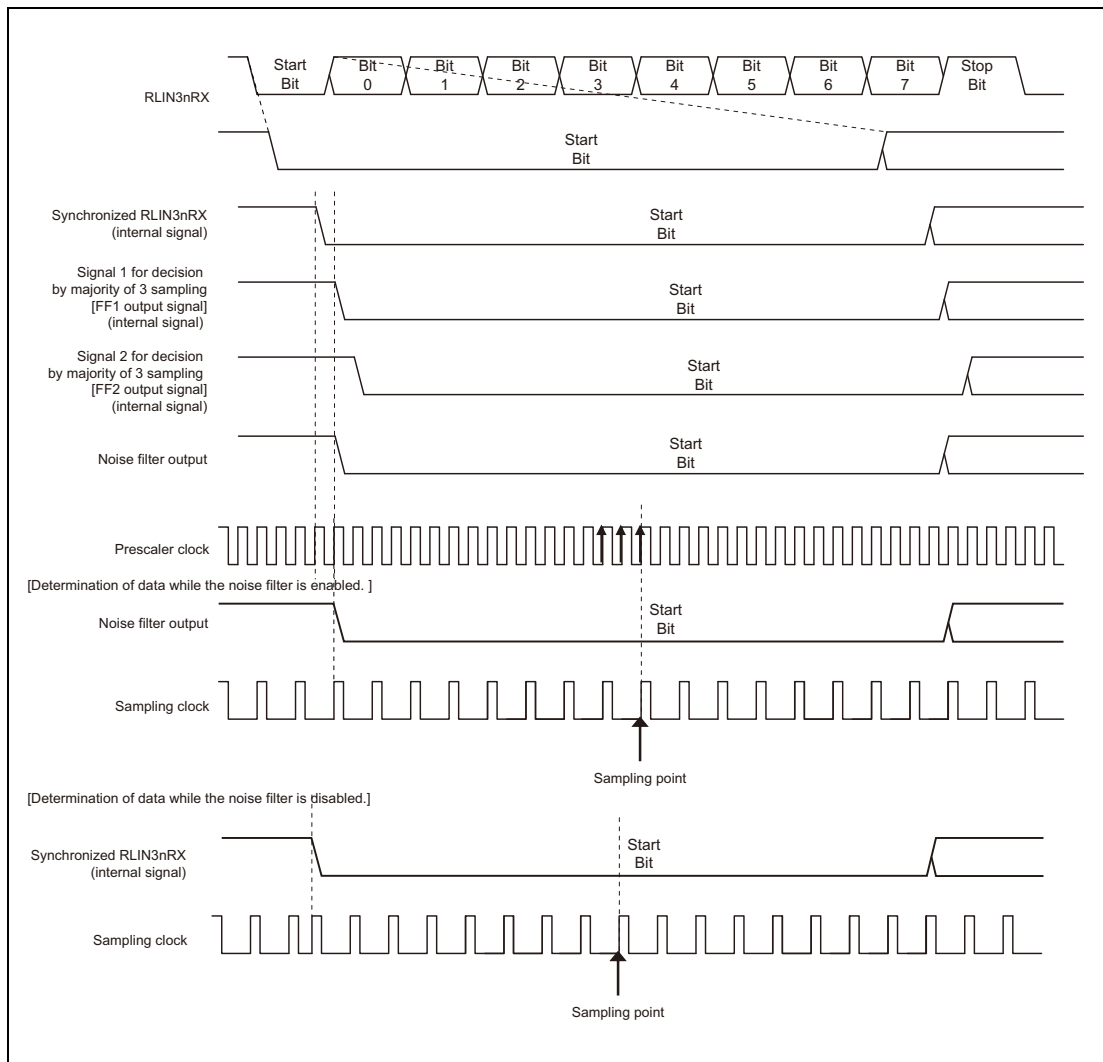


Figure 16.32 Determination of Received Data when Noise Filter is Used

Section 17 CANFD Interface (RS-CANFD)

This section contains a generic description of the CANFD Interface (RS-CANFD).

The first part of this section describes all RH850/P1M-E specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RS-CANFD.

17.1 Features of RH850/P1M-E RS-CANFD

17.1.1 Number of Units and Channels

This microcontroller has the following number of RS-CANFD units.

Table 17.1 Number of Units

| Product Name | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 1 | |
| Name | RSCFDn (n = 0) | |

The individual products have the CANFD Interface Channels listed below.

Table 17.2 Unit Configurations and Channels

| Unit Name | Channel Name | RH850/P1M-E 100 pins | | RH850/P1M-E 144 pins | |
|-----------|--------------|-------------------------|----------------|-------------------------|----------------|
| | | 1 MB (3 ch) | 2 MB (3 ch) | 1 MB (3 ch) | 2 MB (3 ch) |
| RSCFD0 | CAN0 | √ | √ | √ | √ |
| | CAN1 | √ | √ | √ | √ |
| | CAN2 | √ | √ | √ | √ |

The RS-CANFD has two interface modes (classical CAN mode and CAN FD mode) and uses different registers for each mode. There are two types of register names RSCANnXXX and RSCFDnCFDXXX (XXX: arbitrary) depending on interface modes. When explaining specifications common to two registers, register names are described as RSCFDn(CFD)XXX.

Table 17.3 Indices

| Index | Meaning |
|-------|--|
| n | Throughout this section, the individual RS-CANFD units are generically indicated by the index "n" (n = 0); for example, RSCFDn(CFD)GCTR is the global control register of the RSCFDn unit. |
| m | Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index "m" (m = 0 to 2); for example, RSCFDn(CFD)CmSTS is the channel m status register. |
| j | The individual registers associated with receive rule table are generically indicated by the index "j" (j = 0 to 15); for example, RSCFDn(CFD)GAFLIDj is the receive rule ID register. |
| k | The individual transmit/receive FIFO buffers are generically indicated by the index "k" (k = 0 to [channel m × 3 + 2]); for example, RSCFDn(CFD)CFCK is the transmit/receive FIFO buffer configuration/control register. |
| x | The individual receive FIFO buffers are generically identified by the index "x" (x = 0 to 7); for example, RSCFDn(CFD)RFSTx is the receive FIFO buffer status register. |
| d | Data field registers of transmit/receive FIFO buffers and receive FIFO buffers are identified by "d" (classical CAN mode: d = 0 to 1, CAN FD mode: d = 0 to 15). For example, the transmit/receive FIFO buffer data field register is described as RSCFDn(CFD)CFDFd_k. |
| q | The individual receive buffers are generically indicated by the index "q" (q = 0 to [channel m × 16 + 15]); for example, RSCFDn(CFD)RMIDq is the receive buffer ID register. |
| p | The individual transmit buffers are generically indicated by the index "p" (p = 0 to [channel m × 16 + 15]); for example, RSCFDn(CFD)TMCp is the transmit buffer control register. |
| b | Data field registers of receive buffers and transmit buffers are identified by "b" (classical CAN mode: b = 0 to 1, CAN FD mode: b = 0 to 4). For example, the receive buffer data field register is described as RSCFDn(CFD)RMDFb_q. |
| r | The individual RAM tests for CAN are generically indicated by the index "r" (r = 0 to 63); for example, RSCFDn(CFD)RPGACCr is the RAM test page access register. |
| y | The registers not covered above are indicated by the letter "y" (y = 0, 1); for example, RSCFDn(CFD)RMNDy is a receive buffer new data register. |

Note: The functions and descriptions of registers in this section are for the RS-CANFDs that has 3 channels (m = 0 to 2). When referring to information with indices, regard the index values as the ones corresponding to your target product.

The following table lists the values of indices for individual products.

Table 17.4 Indices for Individual Products (1/2)

| Index Correspondence of Each Product | | | | |
|--------------------------------------|---|---|---|---|
| | RH850/P1M-E 100 pins | | RH850/P1M-E 144 pins | |
| | 1 MB (3 ch) | 2 MB (3 ch) | 1 MB (3 ch) | 2 MB (3 ch) |
| j | 0 to 15 | 0 to 15 | 0 to 15 | 0 to 15 |
| k | 0 to 8 | 0 to 8 | 0 to 8 | 0 to 8 |
| x | 0 to 7 | 0 to 7 | 0 to 7 | 0 to 7 |
| d | 0, 1* ¹ 0 to 15* ² | 0, 1* ¹ 0 to 15* ² | 0, 1* ¹ 0 to 15* ² | 0, 1* ¹ 0 to 15* ² |
| q | 0 to 47 | 0 to 47 | 0 to 47 | 0 to 47 |
| p | 0 to 47 | 0 to 47 | 0 to 47 | 0 to 47 |

Table 17.4 Indices for Individual Products (2/2)

| Index Correspondence of Each Product | | | | |
|--------------------------------------|--|--|--|--|
| | RH850/P1M-E 100 pins | | RH850/P1M-E 144 pins | |
| | 1 MB (3 ch) | 2 MB (3 ch) | 1 MB (3 ch) | 2 MB (3 ch) |
| b | 0, 1 ^{*1} 0 to 4 ^{*2} | 0, 1 ^{*1} 0 to 4 ^{*2} | 0, 1 ^{*1} 0 to 4 ^{*2} | 0, 1 ^{*1} 0 to 4 ^{*2} |
| r | 0 to 63 | 0 to 63 | 0 to 63 | 0 to 63 |
| y | 0, 1 | 0, 1 | 0, 1 | 0, 1 |

Note 1. In classical CAN mode

Note 2. In CAN FD mode

17.1.2 Register Base Address

RSCFDn base addresses are listed in the following table.

RSCFDn register addresses are given as offsets from the base addresses in general.

Table 17.5 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <RSCFD0_base> | FFD2 0000 _H |

17.1.3 Clock Supply

The RSCFDn clock supply is shown in the following table.

Table 17.6 Clock Supply

| Unit Name | Clock for the Unit | Supply Clock Name |
|-----------|--------------------|-------------------------------------|
| RSCFDn | clk_xincan | Main OSC clock CLK_MOSC |
| | clk | Low-speed peripheral clock CLK_LSB |
| | pclk | High-speed peripheral clock CLK_HSB |

The operating frequency of the RSCFDn depends on the transfer rate and the number of channels in use. **Table 17.7** shows the range of the frequency.

Table 17.7 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/P1M-E

| Condition | | Range of Operating Frequency | | |
|---------------------------------|------------------------|------------------------------|--------------------------|-------------------|
| Transfer Rate | No. of Channels in Use | pclk | clk_xincan* ¹ | clk* ¹ |
| (Transfer Rate) ≤ 2Mbps | max. 3ch | 80 MHz (fixed) | 16 MHz (fixed) | 40 MHz (fixed) |
| 2Mbps < (Transfer Rate) ≤ 8Mbps | max. 3ch | 80 MHz (fixed) | (do not select) | 40 MHz (fixed) |

Note 1. Setting the DCS bit in the RSCFDn(CFD)GCFG register enables to select either clk_xincan or clk.

17.1.4 Interrupt Requests

The RSCFDn interrupt requests are listed in the following table.

Table 17.8 Interrupt Requests

| Unit Interrupt Signal | Outline | Interrupt Number | DMA Trigger Number |
|-----------------------|----------------------------|---|--|
| RSCFDn | | | |
| INTRCANGERR | CAN global error interrupt | ch189 | — |
| INTRCANGRECC | CAN receive FIFO interrupt | ch190 | — |
| CAN0 | INTRCAN0ERR | CAN0 error interrupt | ch183 |
| | INTRCAN0REC | CAN0 transmit/receive FIFO receive completion interrupt | ch184 |
| | INTRCAN0TRX | CAN0 transmit interrupt | ch185 |
| CAN1 | INTRCAN1ERR | CAN1 error interrupt | ch186 |
| | INTRCAN1REC | CAN1 transmit/receive FIFO receive completion interrupt | ch187 |
| | INTRCAN1TRX | CAN1 transmit interrupt | ch188 |
| CAN2 | INTRCAN2ERR | CAN2 error interrupt | ch191 |
| | INTRCAN2REC | CAN2 transmit/receive FIFO receive completion interrupt | ch192 |
| | INTRCAN2TRX | CAN2 transmit interrupt | ch193 |
| DMACANRF0 | RX FIFO DMA request 0 | — | ch13 ^{*1} or ch34 ^{*2} |
| DMACANRF1 | RX FIFO DMA request 1 | — | ch14 ^{*1} or ch35 ^{*2} |
| DMACANRF2 | RX FIFO DMA request 2 | — | ch15 ^{*1} or ch36 ^{*2} |
| DMACANRF3 | RX FIFO DMA request 3 | — | ch16 ^{*1} or ch37 ^{*2} |
| DMACANRF4 | RX FIFO DMA request 4 | — | ch17 ^{*1} or ch38 ^{*2} |
| DMACANRF5 | RX FIFO DMA request 5 | — | ch18 ^{*1} or ch39 ^{*2} |
| DMACANRF6 | RX FIFO DMA request 6 | — | ch19 ^{*1} or ch40 ^{*2} |
| DMACANRF7 | RX FIFO DMA request 7 | — | ch20 ^{*1} or ch41 ^{*2} |
| DMACANCF0 | COM FIFO DMA request 0 | — | ch21 ^{*1} or ch42 ^{*2} |
| DMACANCF1 | COM FIFO DMA request 1 | — | ch22 ^{*1} or ch43 ^{*2} |
| DMACANCF2 | COM FIFO DMA request 2 | — | ch23 ^{*1} or ch44 ^{*2} |

Note 1. Exclusively used with INTTAUD0li (i = 0 to 10)

Note 2. Exclusively used with INTTAUD1li (i = 0 to 10)

NOTE

For selecting the DMA requests, see **Section 7.11.1, DTS Trigger Select Register Descriptions.**

17.1.5 Reset Source

Please refer to **Section 8, Reset Controller**.

17.1.6 External Input/Output Signals

External input/output signals of RSCFDn are listed below.

Table 17.9 External Input/Output Signals

| Unit Signal Name | Outline | Alternative Port Pin Signal |
|---------------------|---------------------------|-----------------------------|
| RSCFD0 | | |
| CANmRX (m = 0 to 2) | CANm receive data input | CANmRX (m = 0 to 2) |
| CANmTX (m = 0 to 2) | CANm transmit data output | CANmTX (m = 0 to 2) |

17.1.7 Combinations of Pins and Ports

Combinations of RS-CANFD0 pins and ports are listed in the following table.

Table 17.10 Combinations of Pins and Ports

| Function | Channel Name | Pin Name | Port Name | | | |
|----------|--------------|-----------|-----------|---------|---------|---------|
| | | | Group 1 | Group 2 | Group 3 | Group 4 |
| RSCFD0 | CAN0 | RSCAN0RX0 | P2_0 | P3_7 | P4_5 | — |
| | | RSCAN0TX0 | P2_1 | P3_8 | P4_6 | — |
| | CAN1 | RSCAN0RX1 | P2_2 | P3_12 | P4_2 | P4_7*1 |
| | | RSCAN0TX1 | P2_3 | P3_13 | P4_3 | P4_3 |
| | CAN2 | RSCAN0RX2 | P5_6 | P5_6 | — | — |
| | | RSCAN0TX2 | P5_7*1 | P5_5 | — | — |

Note 1. Usable in the 144-pin product

17.2 Overview

17.2.1 Functional Overview

Table 17.11 shows the RS-CANFD module specifications. **Figure 17.1** shows the RS-CANFD module block diagram.

Table 17.11 RS-CANFD Module Specifications (1/3)

| Item | Specification |
|---------------------|--|
| Number of channels | 3 |
| Protocol | ISO11898-1 compliant Using CAN FD frames is selectable by switching interface modes. |
| Communication speed | <p>Classical CAN mode:</p> <ul style="list-style-type: none"> Maximum 1 Mbps $\text{Communication speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ $\text{CANm bit time} = \text{CANmTq} \times \text{Tq count per bit}$ $\text{CANmTq} = \frac{(\text{BRP}[9:0] \text{ bits in the RSCANnCmCFG register} + 1)}{f_{\text{CAN}}}$ <p>f_{CAN}: Frequency of CAN clock (selected by the DCS bit in the RSCANnGCFG register)</p> <p>CAN FD mode:</p> <ul style="list-style-type: none"> Nominal bit rate: max.1 Mbps, data bit rate: max. 8 Mbps $\text{Transmission rate (CANm nominal bit time clock)} = \frac{1}{\text{CANm nominal bit time}}$ $\text{Transmission rate (CANm data bit time clock)} = \frac{1}{\text{CANm data bit time}}$ $\text{CANm nominal bit time} = \text{CANmTq(N)} \times \text{Tq count per nominal bit}$ $\text{CANm data bit time} = \text{CANmTq(D)} \times \text{Tq count per data bit}$ $\text{CANmTq(N)} = \frac{(\text{NBRP}[9:0] \text{ bits in the RSCFDnCFDCmNCFG register} + 1)}{f_{\text{CAN}}}$ $\text{CANmTq(D)} = \frac{(\text{DBRP}[7:0] \text{ bits in the RSCFDnCFDCmDCFG register} + 1)}{f_{\text{CAN}}}$ <p>f_{CAN}: Frequency of CAN clock (selected by the DCS bit in the RSCFDnCFDGCFG register)</p> <p>$m = 0$ to 2 Tq: Time quantum</p> |
| Buffer | <p>240 buffers in total</p> <ul style="list-style-type: none"> Individual buffers: 48 buffers (16 buffers × 3 channels) Transmit buffer: 16 buffers per channel Transmit queue: Single queue per channel (shared with the transmit buffer; up to 16 buffers allocatable) Shared buffers: 192 buffers for all channels Receive buffer: 0 to 48 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) ECC included |
| Reception function | <ul style="list-style-type: none"> Receives data frames and remote frames. Selects ID format (standard ID, extended ID, or both IDs) to be received. Sets interrupt enable/disable for each FIFO. Mirror function (reception of messages transmitted from the own CAN node) Timestamp function (to record message reception time as a 16-bit timer value) |

Table 17.11 RS-CANFD Module Specifications (2/3)

| Item | Specification |
|-----------------------------------|---|
| Reception filter function | <ul style="list-style-type: none"> • Selects receive messages according to 192 receive rules. • Sets the number of receive rules (0 to 128) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Enables DLC filter check for each acceptance rule. |
| Receive message transfer function | <ul style="list-style-type: none"> • Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Label addition function Stores label information together with a message in a receive buffer and FIFO buffer. |
| Transmission function | <ul style="list-style-type: none"> • Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function |
| Interval transmission function | Transmits messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers) |
| Transmit queue function | Transmits all stored messages according to the ID priority. |
| Transmit history function | Stores the history information of transmission-completed messages Adds timestamp (recording message transmission time as a 16-bit timer value) to the history information. |
| Gateway function | Transmits a received message automatically. |
| Bus off recovery mode selection | Selects the method for returning from bus off state. <ul style="list-style-type: none"> • ISO11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Transition to channel halt mode by program request • Transition to the error-active state by program request (forcible return from the bus off state) |
| Error status monitoring | <ul style="list-style-type: none"> • Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors. |
| Interrupt source | 11 sources <ul style="list-style-type: none"> • Global interrupts (2 sources) Receive FIFO interrupt Global error interrupt • Channel interrupts (3 sources/channel) CANm transmit interrupt (m = 0 to 2) <ul style="list-style-type: none"> – CANm transmit complete interrupt – CANm transmit abort interrupt – CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode) – CANm transmit history interrupt – CANm transmit queue interrupt CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) CANm error interrupt |
| CAN stop mode | Reduces power consumption by stopping clock supply to the RS-CANFD module. |
| CAN clock source | Selects the clk or the clk_xincan. As for the range of operating frequency, see Table 17.7 . |

Table 17.11 RS-CANFD Module Specifications (3/3)

| Item | Specification |
|---------------|--|
| Test function | Test function for user evaluation <ul style="list-style-type: none"> • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • Restricted operation mode • RAM test (read/write test) • Inter-channel communication test [CRC error test enabled] |

17.2.2 Interface Modes

The RS-CANFD has two interface modes.

- Classical CAN mode: Handles only classical CAN frames.
- CAN FD mode: Handles classical CAN frames and CAN FD frames.

These two modes use different register maps with the same base address. Register maps change by switching interface modes.

Interface modes are switched by the RCMC bit in the RSCFDn(CFD)GRMCFG register.

17.2.3 Block Diagram

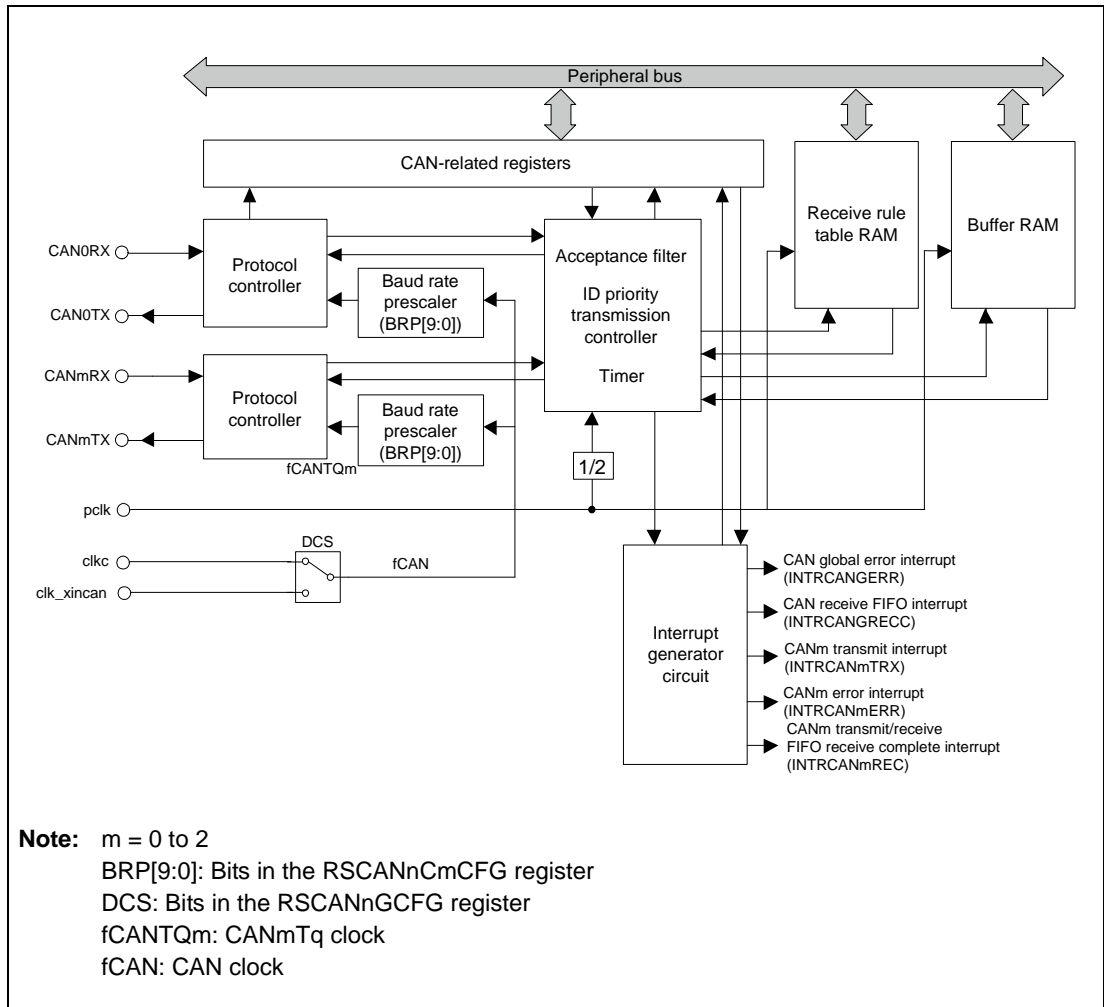


Figure 17.1 RS-CANFD Module Block Diagram (Classical CAN Mode)

In CAN FD mode, different clock signals are input to the baud rate prescaler and the protocol controller respectively. See **Section 17.11.1.3, Communication Speed Setting**.

17.3 Registers (Classical CAN Mode)

17.3.1 List of Registers

The following tables list RS-CANFD registers to be used in classical CAN mode.

For details about <RSCFDn_base>, see **Section 17.1.2, Register Base Address**.

For details about registers initialized in Global reset mode or Channel reset mode, see following.

- **Table 17.180, Registers Initialized in Global Reset Mode or Channel Reset Mode**
- **Table 17.181, Registers Initialized Only in Global Reset Mode**

Table 17.12 Registers (1/2)

| Module | Register | Symbol | Address |
|--|--|-----------------|---|
| Interface mode-related registers | | | |
| RSCANn | Global interface mode select register | RSCANnGRMCFG | <RSCFDn_base> + 04FC _H |
| Channel-related registers | | | |
| RSCANn | Channel m Configuration Register | RSCANnCmCFG | <RSCFDn_base> + 0000 _H + (10 _H × m) |
| RSCANn | Channel m control register | RSCANnCmCTR | <RSCFDn_base> + 0004 _H + (10 _H × m) |
| RSCANn | Channel m status register | RSCANnCmSTS | <RSCFDn_base> + 0008 _H + (10 _H × m) |
| RSCANn | Channel m error flag register | RSCANnCmERFL | <RSCFDn_base> + 000C _H + (10 _H × m) |
| Global-related registers | | | |
| RSCANn | Global configuration register | RSCANnGCFG | <RSCFDn_base> + 0084 _H |
| RSCANn | Global control register | RSCANnGCTR | <RSCFDn_base> + 0088 _H |
| RSCANn | Global status register | RSCANnGSTS | <RSCFDn_base> + 008C _H |
| RSCANn | Global error flag register | RSCANnGERFL | <RSCFDn_base> + 0090 _H |
| RSCANn | Global timestamp counter register | RSCANnGTSC | <RSCFDn_base> + 0094 _H |
| RSCANn | Global TX Interrupt Status Register 0 | RSCANnGTINTSTS0 | <RSCFDn_base> + 0460 _H |
| RSCANn | Global FD configuration register | RSCANnGFDCFG | <RSCFDn_base> + 0474 _H |
| Receive rule-related registers | | | |
| RSCANn | Receive Rule Entry Control Register | RSCANnGAFLECTR | <RSCFDn_base> + 0098 _H |
| RSCANn | Receive Rule Configuration Register 0 | RSCANnGAFLCFG0 | <RSCFDn_base> + 009C _H |
| RSCANn | Receive Rule ID Register j | RSCANnGAFLIDj | <RSCFDn_base> + 0500 _H + (10 _H × j) |
| RSCANn | Receive Rule Mask Register j | RSCANnGAFLMj | <RSCFDn_base> + 0504 _H + (10 _H × j) |
| RSCANn | Receive Rule Pointer 0 Register j | RSCANnGAFLP0_j | <RSCFDn_base> + 0508 _H + (10 _H × j) |
| RSCANn | Receive Rule Pointer 1 Register j | RSCANnGAFLP1_j | <RSCFDn_base> + 050C _H + (10 _H × j) |
| Receive buffer-related registers | | | |
| RSCANn | Receive Buffer Number Register | RSCANnRMNB | <RSCFDn_base> + 00A4 _H |
| RSCANn | Receive Buffer New Data Register y | RSCANnRMNDy | <RSCFDn_base> + 00A8 _H + (04 _H × y) |
| RSCANn | Receive Buffer ID Register q | RSCANnRMIDq | <RSCFDn_base> + 0600 _H + (10 _H × q) |
| RSCANn | Receive Buffer Pointer Register q | RSCANnRMPTRq | <RSCFDn_base> + 0604 _H + (10 _H × q) |
| RSCANn | Receive Buffer Data Field 0 Register q | RSCANnRMDf0_q | <RSCFDn_base> + 0608 _H + (10 _H × q) |
| RSCANn | Receive Buffer Data Field 1 Register q | RSCANnRMDf1_q | <RSCFDn_base> + 060C _H + (10 _H × q) |
| Receive FIFO buffer-related registers | | | |
| RSCANn | Receive FIFO Buffer Configuration and Control Register x | RSCANnRFCCx | <RSCFDn_base> + 00B8 _H + (04 _H × x) |
| RSCANn | Receive FIFO Buffer Status Register x | RSCANnRFSTSx | <RSCFDn_base> + 00D8 _H + (04 _H × x) |
| RSCANn | Receive FIFO Buffer Pointer Control Register x | RSCANnRFPCTRx | <RSCFDn_base> + 00F8 _H + (04 _H × x) |
| RSCANn | Receive FIFO Buffer Access ID Register x | RSCANnRFIDx | <RSCFDn_base> + 0E00 _H + (10 _H × x) |
| RSCANn | Receive FIFO Buffer Access Pointer Register x | RSCANnRFPTRx | <RSCFDn_base> + 0E04 _H + (10 _H × x) |
| RSCANn | Receive FIFO Buffer Access Data Field 0 Register x | RSCANnRFDf0_x | <RSCFDn_base> + 0E08 _H + (10 _H × x) |

Table 17.12 Registers (2/2)

| Module | Register | Symbol | Address |
|---|--|-----------------|---|
| RSCANn | Receive FIFO Buffer Access Data Field 1 Register x | RSCANnRFDF1_x | <RSCFDn_base> + 0E0C _H + (10 _H × x) |
| Transmit/Receive FIFO buffer related registers | | | |
| RSCANn | Transmit/receive FIFO Buffer Configuration and Control Register k | RSCANnCFCCk | <RSCFDn_base> + 0118 _H + (04 _H × k) |
| RSCANn | Transmit/receive FIFO Buffer Status Register k | RSCANnCFSTSk | <RSCFDn_base> + 0178 _H + (04 _H × k) |
| RSCANn | Transmit/receive FIFO Buffer Pointer Control Register k | RSCANnCFPCTRk | <RSCFDn_base> + 01D8 _H + (04 _H × k) |
| RSCANn | Transmit/receive FIFO Buffer Access ID Register k | RSCANnCFIDk | <RSCFDn_base> + 0E80 _H + (10 _H × k) |
| RSCANn | Transmit/receive FIFO Buffer Access Pointer Register k | RSCANnCFPTRk | <RSCFDn_base> + 0E84 _H + (10 _H × k) |
| RSCANn | Transmit/receive FIFO Buffer Access Data Field 0 Register k | RSCANnCFDF0_k | <RSCFDn_base> + 0E88 _H + (10 _H × k) |
| RSCANn | Transmit/receive FIFO Buffer Access Data Field 1 Register k | RSCANnCFDF1_k | <RSCFDn_base> + 0E8C _H + (10 _H × k) |
| FIFO status-related registers | | | |
| RSCANn | FIFO Empty Status Register | RSCANnFESTS | <RSCFDn_base> + 0238 _H |
| RSCANn | FIFO Full Status Register | RSCANnFFSTS | <RSCFDn_base> + 023C _H |
| RSCANn | FIFO Message Lost Status Register | RSCANnFMSTS | <RSCFDn_base> + 0240 _H |
| RSCANn | Receive FIFO Buffer Interrupt Flag Status Register | RSCANnRFISTS | <RSCFDn_base> + 0244 _H |
| RSCANn | Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register | RSCANnCFRISTS | <RSCFDn_base> + 0248 _H |
| RSCANn | Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register | RSCANnCFRISTS | <RSCFDn_base> + 024C _H |
| Transmit buffer-related registers | | | |
| RSCANn | Transmit Buffer Control Register p | RSCANnTMCp | <RSCFDn_base> + 0250 _H + (01 _H × p) |
| RSCANn | Transmit Buffer Status Register p | RSCANnTMSTSp | <RSCFDn_base> + 02D0 _H + (01 _H × p) |
| RSCANn | Transmit Buffer ID Register p | RSCANnTMIDp | <RSCFDn_base> + 1000 _H + (10 _H × p) |
| RSCANn | Transmit Buffer Pointer Register p | RSCANnTMPTRp | <RSCFDn_base> + 1004 _H + (10 _H × p) |
| RSCANn | Transmit Buffer Data Field 0 Register p | RSCANnTMDf0_p | <RSCFDn_base> + 1008 _H + (10 _H × p) |
| RSCANn | Transmit Buffer Data Field 1 Register p | RSCANnTMDf1_p | <RSCFDn_base> + 100C _H + (10 _H × p) |
| RSCANn | Transmit Buffer Interrupt Enable Configuration Register y | RSCANnTMIECy | <RSCFDn_base> + 0390 _H + (04 _H × y) |
| Transmit buffer status-related registers | | | |
| RSCANn | Transmit Buffer Transmit Request Status Register y | RSCANnTMTRSTSy | <RSCFDn_base> + 0350 _H + (04 _H × y) |
| RSCANn | Transmit Buffer Transmit Abort Request Status Register y | RSCANnTMTARSTSy | <RSCFDn_base> + 0360 _H + (04 _H × y) |
| RSCANn | Transmit Buffer Transmit Complete Status Register y | RSCANnTMTCASTSy | <RSCFDn_base> + 0370 _H + (04 _H × y) |
| RSCANn | Transmit Buffer Transmit Abort Status Register y | RSCANnTMTASTSy | <RSCFDn_base> + 0380 _H + (04 _H × y) |
| Transmit queue-related registers | | | |
| RSCANn | Transmit Queue Configuration and Control Register m | RSCANnTXQCCm | <RSCFDn_base> + 03A0 _H + (04 _H × m) |
| RSCANn | Transmit Queue Status Register m | RSCANnTXQSTSm | <RSCFDn_base> + 03C0 _H + (04 _H × m) |
| RSCANn | Transmit Queue Pointer Control Register m | RSCANnTXQPCTRM | <RSCFDn_base> + 03E0 _H + (04 _H × m) |
| Transmit history-related registers | | | |
| RSCANn | Transmit History Configuration and Control Register m | RSCANnTHLCCm | <RSCFDn_base> + 0400 _H + (04 _H × m) |
| RSCANn | Transmit History Status Register m | RSCANnTHLSTSm | <RSCFDn_base> + 0420 _H + (04 _H × m) |
| RSCANn | Transmit History Pointer Control Register m | RSCANnTHLPCTRM | <RSCFDn_base> + 0440 _H + (04 _H × m) |
| RSCANn | Transmit History Access Register m | RSCANnTHLACcm | <RSCFDn_base> + 1800 _H + (04 _H × m) |
| Test-related registers | | | |
| RSCANn | Global Test Configuration Register | RSCANnGTSTCFG | <RSCFDn_base> + 0468 _H |
| RSCANn | Global Test Control Register | RSCANnGTSTCTR | <RSCFDn_base> + 046C _H |
| RSCANn | Global Lock Key Register | RSCANnGLOCKK | <RSCFDn_base> + 047C _H |
| RSCANn | RAM Test Page Access Register r | RSCANnRPGACCr | <RSCFDn_base> + 1900 _H + (04 _H × r) |
| Mode read registers | | | |
| RSCANn | CAN FD Mode Read Register | RSCANnCANFMDR | <RSCFDn_base> + 8000 _H |

Table 17.13 Transmit Buffer p Allocated to Each Channel

| CANm | |
|----------------------|-----------------------------|
| Transmit buffer p | Transmit buffer 16 × m + 0 |
| | Transmit buffer 16 × m + 1 |
| | Transmit buffer 16 × m + 2 |
| | Transmit buffer 16 × m + 3 |
| | Transmit buffer 16 × m + 4 |
| | Transmit buffer 16 × m + 5 |
| | Transmit buffer 16 × m + 6 |
| | Transmit buffer 16 × m + 7 |
| | Transmit buffer 16 × m + 8 |
| | Transmit buffer 16 × m + 9 |
| | Transmit buffer 16 × m + 10 |
| | Transmit buffer 16 × m + 11 |
| | Transmit buffer 16 × m + 12 |
| | Transmit buffer 16 × m + 13 |
| | Transmit buffer 16 × m + 14 |
| | Transmit buffer 16 × m + 15 |

Table 17.14 Transmit/Receive FIFO Buffer k Allocated to Each Channel

| CANm | |
|-----------------------------------|--|
| Transmit/receive FIFO buffer k | Transmit/receive FIFO buffer 3 × m + 0 |
| | Transmit/receive FIFO buffer 3 × m + 1 |
| | Transmit/receive FIFO buffer 3 × m + 2 |

Table 17.15 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

| Setting of Bits CFTML[3:0] | Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer |
|----------------------------|--|
| 0000 _B | Transmit buffer 16 × m + 0 |
| 0001 _B | Transmit buffer 16 × m + 1 |
| 0010 _B | Transmit buffer 16 × m + 2 |
| 0011 _B | Transmit buffer 16 × m + 3 |
| 0100 _B | Transmit buffer 16 × m + 4 |
| 0101 _B | Transmit buffer 16 × m + 5 |
| 0110 _B | Transmit buffer 16 × m + 6 |
| 0111 _B | Transmit buffer 16 × m + 7 |
| 1000 _B | Transmit buffer 16 × m + 8 |
| 1001 _B | Transmit buffer 16 × m + 9 |
| 1010 _B | Transmit buffer 16 × m + 10 |
| 1011 _B | Transmit buffer 16 × m + 11 |
| 1100 _B | Transmit buffer 16 × m + 12 |
| 1101 _B | Transmit buffer 16 × m + 13 |
| 1110 _B | Transmit buffer 16 × m + 14 |
| 1111 _B | Transmit buffer 16 × m + 15 |

Table 17.16 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

| Setting of Bits TXQDC[3:0] | Transmit Buffer p Allocated to the Transmit Queue |
|----------------------------|--|
| 0000 _B | Setting prohibited |
| 0001 _B | Setting prohibited |
| 0010 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 13$ |
| 0011 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 12$ |
| 0100 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 11$ |
| 0101 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 10$ |
| 0110 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 9$ |
| 0111 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 8$ |
| 1000 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 7$ |
| 1001 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 6$ |
| 1010 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 5$ |
| 1011 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 4$ |
| 1100 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 3$ |
| 1101 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 2$ |
| 1110 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 1$ |
| 1111 _B | Transmit buffer $16 \times m + 15$ to $16 \times m + 0$ |

17.3.2 Details of Interface Mode-Related Registers

17.3.2.1 RSCANnGRMCFG — Global Interface Mode Select Register

Access: RSCANnGRMCFG register can be read/written in 32-bit units
 RSCANnGRMCFG, RSCANnGRMCFGH registers can be read/written in 16-bit units
 RSCANnGRMCFG, RSCANnGRMCFGH, RSCANnGRMCFGH registers can be read/written in 8-bit units

Address: RSCANnGRMCFG: <RSCFDn_base> + 04FC_H
 RSCANnGRMCFG: <RSCFDn_base> + 04FC_H,
 RSCANnGRMCFGH: <RSCFDn_base> + 04FE_H
 RSCANnGRMCFG: <RSCFDn_base> + 04FC_H,
 RSCANnGRMCFGH: <RSCFDn_base> + 04FD_H,
 RSCANnGRMCFGH: <RSCFDn_base> + 04FE_H,
 RSCANnGRMCFGH: <RSCFDn_base> + 04FF_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RCMC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 17.17 RSCANnGRMCFG Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 0 | RCMC | Interface Mode Select 0: Classical CAN mode 1: CAN FD mode |

Note 1. The RSCANnGRMCFG register and the RSCFDnCFDGRMCFG register are the same register. Therefore, set either one of the registers.

Modify the RSCANnGRMCFG register only in global reset mode. Before setting other RS-CANFD registers, set this register.

RCMC Bit

Setting this bit to 0 makes classical CAN mode available. Setting this bit to 1 causes the RS-CANFD module to transition to CAN FD mode. To switch the RS-CANFD module from CAN FD mode to classical CAN mode, set the values after reset to all respective registers and bits allocated only to the register map in CAN FD mode, and then modify the value of RSCANnGRMCFG register.

17.3.3 Details of Channel-Related Registers

17.3.3.1 RSCANnCmCFG — Channel m Configuration Register (m = 0 to 2)

Access: RSCANnCmCFG register can be read/written in 32-bit units
 RSCANnCmCFGL, RSCANnCmCFGH registers can be read/written in 16-bit units
 RSCANnCmCFGLL, RSCANnCmCFGLH, RSCANnCmCFGHL, RSCANnCmCFGHH registers can be read/
 written in 8-bit units

Address: RSCANnCmCFG: $\langle \text{RSCFDn_base} \rangle + 0000_{\text{H}} + (10_{\text{H}} \times m)$
 RSCANnCmCFGL: $\langle \text{RSCFDn_base} \rangle + 0000_{\text{H}} + (10_{\text{H}} \times m)$,
 RSCANnCmCFGH: $\langle \text{RSCFDn_base} \rangle + 0002_{\text{H}} + (10_{\text{H}} \times m)$
 RSCANnCmCFGLL: $\langle \text{RSCFDn_base} \rangle + 0000_{\text{H}} + (10_{\text{H}} \times m)$,
 RSCANnCmCFGLH: $\langle \text{RSCFDn_base} \rangle + 0001_{\text{H}} + (10_{\text{H}} \times m)$,
 RSCANnCmCFGHL: $\langle \text{RSCFDn_base} \rangle + 0002_{\text{H}} + (10_{\text{H}} \times m)$,
 RSCANnCmCFGHH: $\langle \text{RSCFDn_base} \rangle + 0003_{\text{H}} + (10_{\text{H}} \times m)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----------|-----|-----|------------|-----|-----|------------|-----|-----|-----|
| | — | — | — | — | — | — | SJW[1:0] | | — | TSEG2[2:0] | | | TSEG1[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | BRP[9:0] | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.18 RSCANnCmCFG_Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 26 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 25, 24 | SJW[1:0] | Resynchronization Jump Width Control $b_{25} \ b_{24}$ 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq |
| 23 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 22 to 20 | TSEG2[2:0] | Time Segment 2 Control $b_{22} \ b_{21} \ b_{20}$ 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq |

Table 17.18 RSCANnCMCFG_Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 19 to 16 | TSEG1[3:0] | Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq |
| 15 to 10 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 9 to 0 | BRP[9:0] | Prescaler Division Ratio setting When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1. |

Modify the RSCANnCMCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode before transitioning to channel communication mode or channel halt mode. For a description of the bit timing parameters and settings, see **Section 17.11.1, Initial Settings**.

SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Allowed values are 1 Tq to 4 Tq, inclusive.

Set a value less than or equal to the value of the TSEG2 bits.

TSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2).

Allowed values are 2 Tq to 8 Tq, inclusive.

Set a value smaller than the value of the TSEG1 bits.

TSEG1[3:0] Bits

These bits are used to specify a Tq value for the total length of the propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1).

Allowed values are 4 Tq to 16 Tq, inclusive.

BRP[9:0] Bits

The CANmTq clock (fCANTQm) is calculated by dividing the CAN clock (fCAN) by the baud rate prescaler, ((BRP[9:0]) + 1). One clock cycle of the CANmTq clock is 1 Time Quantum (Tq).

17.3.3.2 RSCANnCMCTR — Channel m Control Register (m = 0 to 2)

Access: RSCANnCMCTR register can be read/written in 32-bit units
 RSCANnCMCTRL, RSCANnCMCTRH registers can be read/written in 16-bit units
 RSCANnCMCTRL, RSCANnCMCTRLH, RSCANnCMCTRHL, RSCANnCMCTRHH registers can be read/
 written in 8-bit units

Address: RSCANnCMCTR: <RSCFDn_base> + 0004_H + (10_H × m)
 RSCANnCMCTRL: <RSCFDn_base> + 0004_H + (10_H × m),
 RSCANnCMCTRH: <RSCFDn_base> + 0006_H + (10_H × m)
 RSCANnCMCTRL: <RSCFDn_base> + 0004_H + (10_H × m),
 RSCANnCMCTRLH: <RSCFDn_base> + 0005_H + (10_H × m),
 RSCANnCMCTRHL: <RSCFDn_base> + 0006_H + (10_H × m),
 RSCANnCMCTRHH: <RSCFDn_base> + 0007_H + (10_H × m)

Value after reset: 0000 0005_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|------|------|-------|-------|-----------|------|------|----------|-----|-----|----|------|-------|------------|------|
| | — | CRCT | — | — | — | CTMS[1:0] | CTME | ERRD | BOM[1:0] | — | — | — | — | — | — | TAIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ALIE | BLIE | OLIE | BORIE | BOEIE | EPIE | EWIE | BEIE | — | — | — | — | RTBO | CSLPR | CHMDC[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 17.19 RSCANnCMCTR Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 30 | CRCT | CRC Error Test Enable 0: The first bit of the reception ID field is not inverted. 1: The first bit of the reception ID field is inverted. |
| 29 to 27 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 26, 25 | CTMS[1:0] | Communication Test Mode Select b ₂₆ b ₂₅ 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode) |
| 24 | CTME | Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled. |
| 23 | ERRD | Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in the RSCANnCMERFL register are all cleared. 1: Error flags for all error information are displayed. |
| 22, 21 | BOM[1:0] | Bus Off Recovery Mode Select b ₂₂ b ₂₁ 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (in bus-off state) by program request |
| 20 to 17 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 16 | TAIE | Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled. |

Table 17.19 RSCANnCMCTR Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 15 | ALIE | Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled. |
| 14 | BLIE | Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled. |
| 13 | OLIE | Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled. |
| 12 | BORIE | Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled. |
| 11 | BOEIE | Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled. |
| 10 | EPIE | Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled. |
| 9 | EWIE | Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled. |
| 8 | BEIE | Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled. |
| 7 to 4 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 3 | RTBO | Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0. |
| 2 | CSLPR | Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode |
| 1, 0 | CHMDC[1:0] | Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited |

CRCT Bit

This bit is used to test the CRC generation circuit in the RS-CANFD module. Setting this bit to 1 inverts the first bit of the ID field when a message is received. With this inversion of bit, the CRC calculation result does not match the normal CRC value of the received frame, which can detect a CRC error (the CERR bit in the RSCANnCMERFL register is 1). When using this function, note the following.

- This function is available while the CTME bit in the RSCANnCMCTR register is 1 (communication test mode enabled).
- This function cannot communicate with other CAN nodes. Use this function for inter-channel communication test (the CmICBCE bit in the RSCANnGTSTCFG register is 1).

- Bit inversion in the ID field may cause bit stuffing rule violation. In that case, no CRC error is detected but a stuff error is detected.

Modify this bit only in channel halt mode. This bit is always 0 in channel reset mode.

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify this bit in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCANnCMERFL register.

When this bit is cleared to 0, if any error is detected while the flags of bits 14-8 in the RSCANnCMERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CANFD module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CANFD module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CANFD module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCANnCMCTR register (m = 0 to 2) are set to 10_B and the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RSCANnCMSTS register are cleared to 00_H.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CANFD module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CANFD module is in the bus off state, the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CANFD module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bits at the same time as the RS-CANFD module transitions to channel halt mode (at bus off entry when the BOM[1:0] bits are 01_B or at bus off end when the BOM[1:0] bits are 10_B), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCANnCMERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCANnCMERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLV flag in the RSCANnCMERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCANnCMERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCANnCMERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCANnCMERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCANnCMERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCANnCMERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly changes the state from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCANn CmSTS register to 00_H and also clears the BOSTS flag in the RSCANn CmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCANn CmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RSCAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.

Clearing this bit to 0 makes the channel exit channel stop mode.

This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **Section 17.6.2, Channel Modes**. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B.

When the RSCAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits are automatically set to 10_B.

17.3.3.3 RSCANn CmSTS — Channel m Status Register (m = 0 to 2)

Access: RSCANn CmSTS register can be read only in 32-bit units
 RSCANn CmSTSL, RSCANn CmSTSH registers can be read only in 16-bit units
 RSCANn CmSTSLL, RSCANn CmSTSLH, RSCANn CmSTSHL, RSCANn CmSTSHH registers can be read only in 8-bit units

Address: RSCANn CmSTS: <RSCFDn_base> + 0008_H + (10_H × m)
 RSCANn CmSTSL: <RSCFDn_base> + 0008_H + (10_H × m),
 RSCANn CmSTSH: <RSCFDn_base> + 000A_H + (10_H × m)
 RSCANn CmSTSLL: <RSCFDn_base> + 0008_H + (10_H × m),
 RSCANn CmSTSLH: <RSCFDn_base> + 0009_H + (10_H × m),
 RSCANn CmSTSHL: <RSCFDn_base> + 000A_H + (10_H × m),
 RSCANn CmSTSHH: <RSCFDn_base> + 000B_H + (10_H × m)

Value after reset: 0000 0005_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|----|----|----|----|----|----|----|----------|--------|--------|-------|-------|---------|---------|---------|
| | TEC[7:0] | | | | | | | | REC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | COMSTS | RECSTS | TRMSTS | BOSTS | EPSTS | CSLPSTS | CHLTSTS | CRSTSTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.20 RSCANn CmSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 24 | TEC[7:0] | The transmit error counter (TEC) can be read. |
| 23 to 16 | REC[7:0] | The receive error counter (REC) can be read. |
| 15 to 8 | Reserved | These bits are read as the value after reset. |
| 7 | COMSTS | Communication Status Flag 0: Communication is not ready. 1: Communication is ready. |
| 6 | RECSTS | Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception |
| 5 | TRMSTS | Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state |
| 4 | BOSTS | Bus Off Status Flag 0: Not in bus off state 1: In bus off state |
| 3 | EPSTS | Error Passive Status Flag 0: Not in error passive state 1: In error passive state |
| 2 | CSLPSTS | Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode |
| 1 | CHLTSTS | Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode |
| 0 | CRSTSTS | Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode |

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag is set to 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state ($TEC[7:0] > 255$) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CANFD module has entered the error passive state ($(128 \leq TEC[7:0] \leq 255)$ or $(128 \leq REC[7:0])$), It is cleared to 0 when the RS-CANFD module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

17.3.3.4 RSCANnCmERFL — Channel m Error Flag Register (m = 0 to 2)

Access: RSCANnCmERFL register can be read/written in 32-bit units
 RSCANnCmERFLL, RSCANnCmERFLH registers can be read/written in 16-bit units
 RSCANnCmERFLLL, RSCANnCmERFLLH, RSCANnCmERFLHL, RSCANnCmERFLHH registers can be read/written in 8-bit units

Address: RSCANnCmERFL: <RSCFDn_base> + 000C_H + (10_H × m)
 RSCANnCmERFLL: <RSCFDn_base> + 000C_H + (10_H × m),
 RSCANnCmERFLH: <RSCFDn_base> + 000E_H + (10_H × m)
 RSCANnCmERFLLL: <RSCFDn_base> + 000C_H + (10_H × m),
 RSCANnCmERFLLH: <RSCFDn_base> + 000D_H + (10_H × m),
 RSCANnCmERFLHL: <RSCFDn_base> + 000E_H + (10_H × m),
 RSCANnCmERFLHH: <RSCFDn_base> + 000F_H + (10_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CRCREG[14:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | ADERR | BOERR | B1ERR | CERR | AERR | FERR | SERR | ALF | BLF | OVLf | BORF | BOEF | EPF | EWf | BEF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.21 RSCANnCmERFL Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 30 to 16 | CRCREG[14:0] | CRC Calculation Data A CRC value calculated based on the transmit message or receive message is indicated. |
| 15 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 14 | ADERR | ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected. |
| 13 | BOERR | Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected. |
| 12 | B1ERR | Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected. |
| 11 | CERR | CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected. |
| 10 | AERR | ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected. |
| 9 | FERR | Form Error Flag 0: No form error is detected. 1: Form error is detected. |

Table 17.21 RSCANnCMERFL Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 8 | SERR | Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected. |
| 7 | ALF | Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected. |
| 6 | BLF | Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected. |
| 5 | OVLf | Overload Flag 0: No overload is detected. 1: Overload is detected. |
| 4 | BORF | Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected. |
| 3 | BOEF | Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected. |
| 2 | EPF | Error Passive Flag 0: No error passive is detected. 1: Error passive is detected. |
| 1 | EWf | Error Warning Flag 0: No error warning is detected. 1: Error warning is detected. |
| 0 | BEF | Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected. |

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0.

If the ERRD bit in the RSCANnCMCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCANnCMERFL is detected, the flag bits are only set to 1 by the error event if bits 14 to 8 were all 0 at the time when the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCANnCMCTR register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0. This bit is always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected even though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected even though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, a bus lock can be detected again when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLf Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCANnCMCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCANnCMCTR register is set to 1 (forcible return from the bus off state is made).
- The BOM[1:0] bits in the RSCANnCMCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCANnCMCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is entered (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is entered when the BOM[1:0] bits in the RSCANn CmCTR register (m = 0 to 2) are set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag is set to 1 when the error passive state is entered ($(128 \leq \text{TEC}[7:0] \leq 255)$ or $(128 \leq \text{REC}[7:0])$).

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWf Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, BOERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCANn CmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write "0" to the given flag and "1" to the other flags.

17.3.4 Details of Global-Related Registers

17.3.4.1 RSCANnGCFG — Global Configuration Register

Access: RSCANnGCFG register can be read/written in 32-bit units
 RSCANnGCFGL, RSCANnGCFGH registers can be read/written in 16-bit units
 RSCANnGCFGLL, RSCANnGCFGLH, RSCANnGCFGHL, RSCANnGCFGHH registers can be read/written in 8-bit units

Address: RSCANnGCFG: <RSCFDn_base> + 0084_H
 RSCANnGCFGL: <RSCFDn_base> + 0084_H, RSCANnGCFGH: <RSCFDn_base> + 0086_H
 RSCANnGCFGLL: <RSCFDn_base> + 0084_H, RSCANnGCFGLH: <RSCFDn_base> + 0085_H,
 RSCANnGCFGHL: <RSCFDn_base> + 0086_H, RSCANnGCFGHH: <RSCFDn_base> + 0087_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|------|----------|-----|-----|-----|------------|------|-----|-----|-----|-----|-----|------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | ITRCP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSBTCS[2:0] | | TSSS | TSP[3:0] | | | | TMTSC E | EEFE | — | DCS | MME | DRE | DCE | TPRI | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 17.22 RSCANnGCFG Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 16 | ITRCP[15:0] | Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000 _H is prohibited when the interval timer is in use. |
| 15 to 13 | TSBTCS[2:0] | Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 bit time clock 0 0 1: Channel 1 bit time clock 0 1 0: Channel 2 bit time clock 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited |
| 12 | TSSS | Timestamp Source Select 0: pclk/2 ^{*1} 1: Bit time clock |

Table 17.22 RSCANnGCFG Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 11 to 8 | TSP[3:0] | Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768 |
| 7 | TMTSCE | Transmission Timestamp Enable 0: Transmission timestamp is disabled. 1: Transmission timestamp is enabled. |
| 6 | EEFE | ECC Error Flag Enable 0: The ECC error flag is disabled. 1: The ECC error flag is enabled. |
| 5 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 4 | DCS | CAN Clock Source Select*2 0: clkc 1: clk_xincan |
| 3 | MME | Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled. |
| 2 | DRE | DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled. |
| 1 | DCE | DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled. |
| 0 | TPRI | Transmit Priority Select 0: ID priority 1: Transmit buffer number priority |

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Note 2. For the CAN clock frequency settings, see **Table 17.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/P1M-E**.

Modify the RSCANnGCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See **Section 17.8.3.1, Interval Transmission Function**.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the bit time clock that will be the clock source of the timestamp counter.

TSSS Bit

This bit is used to select a clock source of the timestamp counter.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

TMTSCE Bit

Setting this bit to 1 makes it possible to store the timestamp of a transmitted message in the transmit history buffer. The timestamp is stored in TMTS[15:0] bits in the RSCANnTHLACCm register.

EEFE Bit

Setting this bit to 1 sets the EEFE bit in the RSCANnGERFL register to 1 when a 2-bit ECC error is detected during the transmission priority determination. At this time, the message in which a 2-bit ECC error was detected is not transmitted.

DCS Bit

When this bit is set to 0, clk is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN).

For the CAN clock frequency settings, see **Table 17.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/P1M-E**.

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte that exceeds the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCANnGAFLP0_j register to 0000_B before clearing the DCE bit in the RSCANnGCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the transmit buffer with the smallest number has the highest priority.

When using the transmit queue, this bit should be set to 0.

17.3.4.2 RSCANnGCTR — Global Control Register

Access: RSCANnGCTR register can be read/written in 32-bit units
 RSCANnGCTRL, RSCANnGCTRH registers can be read/written in 16-bit units
 RSCANnGCTRL, RSCANnGCTRLH, RSCANnGCTRHL, RSCANnGCTRHH registers can be read/written in 8-bit units

Address: RSCANnGCTR: <RSCFDn_base> + 0088_H
 RSCANnGCTRL: <RSCFDn_base> + 0088_H, RSCANnGCTRH: <RSCFDn_base> + 008A_H
 RSCANnGCTRL: <RSCFDn_base> + 0088_H, RSCANnGCTRLH: <RSCFDn_base> + 0089_H,
 RSCANnGCTRHL: <RSCFDn_base> + 008A_H, RSCANnGCTRHH: <RSCFDn_base> + 008B_H

Value after reset: 0000 0005_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSRST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|--------|------|------|---|---|---|---|---|-------|-----------|-----|
| | — | — | — | — | — | THLEIE | MEIE | DEIE | — | — | — | — | — | GSLPR | GMDC[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 17.23 RSCANnGCTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 17 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 16 | TSRST | Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0. |
| 15 to 11 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 10 | THLEIE | Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled. |
| 9 | MEIE | FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled. |
| 8 | DEIE | DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled. |
| 7 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | GSLPR | Global Stop Mode 0: Other than global stop mode 1: Global stop mode |
| 1, 0 | GMDC[1:0] | Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited |

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCANnGTSC register is cleared to 0000_H.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RSCAN module into global stop mode.

Clearing this bit to 0 release the RSCAN module from global stop mode.

This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CANFD module (global operating mode, global reset mode, or global test mode). For details, see **Section 17.6.1, Global Modes**. Setting the GSLPR bit to 1 when in global reset mode places the RS-CANFD module in global stop mode.

17.3.4.3 RSCANnGSTS — Global Status Register

Access: RSCANnGSTS register can be read only in 32-bit units
 RSCANnGSTSL, RSCANnGSTSH registers can be read only in 16-bit units
 RSCANnGSTSLL, RSCANnGSTSLH, RSCANnGSTSHL, RSCANnGSTSHH registers can be read only in 8-bit units

Address: RSCANnGSTS: <RSCFDn_base> + 008C_H
 RSCANnGSTSL: <RSCFDn_base> + 008C_H, RSCANnGSTSH: <RSCFDn_base> + 008E_H
 RSCANnGSTSLL: <RSCFDn_base> + 008C_H, RSCANnGSTSLH: <RSCFDn_base> + 008D_H,
 RSCANnGSTSHL: <RSCFDn_base> + 008E_H, RSCANnGSTSHH: <RSCFDn_base> + 008F_H

Value after reset: 0000 000D_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|-------------|-------------|------------|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | GRAM NIT | GSLPS TS | GHLT TS | GRST TS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.24 RSCANnGSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 4 | Reserved | When read, the value after reset is returned. |
| 3 | GRAMINIT | CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing. |
| 2 | GSLPSTS | Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode |
| 1 | GHLTSTS | Global Test Status Flag 0: Not in global test mode 1: In global test mode |
| 0 | GRSTSTS | Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode |

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

17.3.4.4 RSCANnGERFL — Global Error Flag Register

Access: RSCANnGERFL register can be read/written in 32-bit units
 RSCANnGERFLL, RSCANnGERFLH registers can be read/written in 16-bit units
 RSCANnGERFLLL, RSCANnGERFLLH, RSCANnGERFLHL, RSCANnGERFLHH registers can be read/
 written in 8-bit units

Address: RSCANnGERFL: <RSCFDn_base> + 0090_H
 RSCANnGERFLL: <RSCFDn_base> + 0090_H, RSCANnGERFLH: <RSCFDn_base> + 0092_H
 RSCANnGERFLLL: <RSCFDn_base> + 0090_H, RSCANnGERFLLH: <RSCFDn_base> + 0091_H,
 RSCANnGERFLHL: <RSCFDn_base> + 0092_H, RSCANnGERFLHH: <RSCFDn_base> + 0093_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | EEF2 | EEF1 | EEF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R/W*1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|-------|-----|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | THLES | MES | DEF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.25 RSCANnGERFL Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 19 | Reserved | When read, the value after reset is returned. When writing these bits, write the value after reset. |
| 18 | EEF2 | ECC Error Flag for Channel 2 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority |
| 17 | EEF1 | ECC Error Flag for Channel 1 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority |
| 16 | EEF0 | ECC Error Flag for Channel 0 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority |
| 15, 14 | Reserved | When read, the value after reset is returned. When writing these bits, write the value after reset. |
| 13 to 8 | Reserved | When read, the undefined value is returned. When writing these bits, write the value after reset. |
| 7, 6 | Reserved | When read, the value after reset is returned. When writing these bits, write the value after reset. |
| 5 | Reserved | When read, the undefined value is returned. When writing this bit, write the value after reset. |
| 4, 3 | Reserved | When read, the value after reset is returned. When writing these bits, write the value after reset. |
| 2 | THLES | Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred. |
| 1 | MES | FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred. |

Table 17.25 RSCANnGERFL Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 0 | DEF | DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred. |

All flags in the RSCANnGERFL register are cleared to 0 in global reset mode.

EEFm Flag

While the EEFE bit in the RSCANnGCFG register is 1, when a 2-bit ECC error is detected during the transmission priority determination of channel m (m = 0 to 2), the EEFm flag is set to 1, disabling message transmission. This flag can be cleared to 0 by writing 0 by the program.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCANnTHLSTSm register (m = 0 to 5) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCANnRFSTSc register (x = 0 to 7) or the CFMLT flags in the RSCANnCFSTSk register (k = 0 to 8) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

NOTE

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to "0" to "0", and the bits not to be set to "0" to "1".

17.3.4.5 RSCANnGTSC — Global Timestamp Counter Register

Access: RSCANnGTSC register can be read only in 32-bit units.
RSCANnGTSL, RSCANnGTSCH registers can be read only in 16-bit units.

Address: RSCANnGTSC: <RSCFDn_base> + 0094_H
RSCANnGTSL: <RSCFDn_base> + 0094_H, RSCANnGTSCH: <RSCFDn_base> + 0096_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.26 RSCANnGTSC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | Reserved | These bits are read as the value after reset. |
| 15 to 0 | TS[15:0] | Timestamp Value The timestamp counter value can be read. Counter Value: 0000 _H to FFFF _H |

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer.

Furthermore, while the TMTSCE bit in the RSCANnGCFG register is 1, the TS[15:0] value is stored in the transmit history buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter start timing and stop timing depend on the count source.

- When the TSSS bit in the RSCANnGCFG register is 0 (pclk):
The timestamp counter starts counting when the RSCAN module has transitioned to global operating mode.
This counter stops counting when the RSCAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm bit time clock):
The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

17.3.4.6 RSCANnGTINTSTS0 — Global TX Interrupt Status Register 0

Access: RSCANnGTINTSTS0 register can be read only in 32-bit units
 RSCANnGTINTSTS0L, RSCANnGTINTSTS0H registers can be read only in 16-bit units
 RSCANnGTINTSTS0LL, RSCANnGTINTSTS0LH, RSCANnGTINTSTS0HL, RSCANnGTINTSTS0HH registers can be read only in 8-bit units

Address: RSCANnGTINTSTS0: <RSCFDn_base> + 0460_H
 RSCANnGTINTSTS0L: <RSCFDn_base> + 0460_H, RSCANnGTINTSTS0H: <RSCFDn_base> + 0462_H
 RSCANnGTINTSTS0LL: <RSCFDn_base> + 0460_H, RSCANnGTINTSTS0LH: <RSCFDn_base> + 0461_H,
 RSCANnGTINTSTS0HL: <RSCFDn_base> + 0462_H, RSCANnGTINTSTS0HH: <RSCFDn_base> + 0463_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|-----------------|-----------------|-----------------|-----------------|-----------------|----|----|----|-----------------|-----------------|-----------------|-----------------|-----------------|
| | — | — | — | — | — | — | — | — | — | — | — | THIF2 | CFTIF2 | TQIF2 | TAIF2 | TSIF2 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | THIF1 | CFTIF1 | TQIF1 | TAIF1 | TSIF1 | — | — | — | THIF0 | CFTIF0 | TQIF0 | TAIF0 | TSIF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R | R | R | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} |

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 17.27 RSCANnGTINTSTS0 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 21 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 20 | THIF2 | Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested. |
| 19 | CFTIF2 | Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested. |
| 18 | TQIF2 | Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested. |
| 17 | TAIF2 | Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested. |
| 16 | TSIF2 | Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested. |
| 15 to 13 | Reserved | These bits are read as the value after reset. |
| 12 | THIF1 | Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested. |
| 11 | CFTIF1 | Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested. |
| 10 | TQIF1 | Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested. |
| 9 | TAIF1 | Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested. |

Table 17.27 RSCANnGTINTSTS0 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 8 | TSIF1 | Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested. |
| 7 to 5 | Reserved | These bits are read as the value after reset. |
| 4 | THIF0 | Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested. |
| 3 | CFTIF0 | Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested. |
| 2 | TQIF0 | Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested. |
| 1 | TAIF0 | Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested. |
| 0 | TSIF0 | Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested. |

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCANnTMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCANnTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCANnCmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCANnTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCANnTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCANnTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCANnTXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCANnCFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCANnCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCANnTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCANnTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCANnTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

17.3.4.7 RSCANnGFDCFG — Global FD Configuration Register

Access: RSCANnGFDCFG register can be read/written in 32-bit unit
 RSCANnGFDCFGL and RSCANnGFDCFGH registers can be read/written in 16-bit unit
 RSCANnGFDCFGLL, RSCANnGFDCFGLH, RSCANnGFDCFGHL, RSCANnGFDCFGHH registers can be read/written in 8-bit unit

Address: RSCANnGFDCFG: <RSCFDn_base> + 0474_H
 RSCANnGFDCFGL: <RSCFDn_base> + 0474_H, RSCANnGFDCFGH: <RSCFDn_base> + 0476_H
 RSCANnGFDCFGLL: <RSCFDn_base> + 0474_H, RSCANnGFDCFGLH: <RSCFDn_base> + 0475_H,
 RSCANnGFDCFGHL: <RSCFDn_base> + 0476_H, RSCANnGFDCFGHH: <RSCFDn_base> + 0477_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|-----|-----|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | TSCCFG[1:0] | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.28 RSCANnGFDCFG Register contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 10 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 9, 8 | TSCCFG[1:0] | Time-stamp capture setting bit b9 b8 0 0: Captured at a sample point in the SOF bit. 0 1: Captured when a valid frame has been transmitted/received. 1 0: Setting prohibited 1 1: Setting prohibited |
| 7 to 1 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 0 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |

TSCCFG bit

Select a point where a time-stamp value is captured. Modify this bit only in global reset mode.

17.3.5 Details of Receive Rule-related Registers

17.3.5.1 RSCANnGAFLECTR — Receive Rule Entry Control Register

Access: RSCANnGAFLECTR register can be read/written in 32-bit units
RSCANnGAFLECTRL, RSCANnGAFLECTRH registers can be read/written in 16-bit units
RSCANnGAFLECTRLL, RSCANnGAFLECTRLH, RSCANnGAFLECTRHL, RSCANnGAFLECTRHH registers can be read/written in 8-bit units

Address: RSCANnGAFLECTR: <RSCFDn_base> + 0098_H
RSCANnGAFLECTRL: <RSCFDn_base> + 0098_H, RSCANnGAFLECTRH: <RSCFDn_base> + 009A_H
RSCANnGAFLECTRLL: <RSCFDn_base> + 0098_H, RSCANnGAFLECTRLH: <RSCFDn_base> + 0099_H,
RSCANnGAFLECTRHL: <RSCFDn_base> + 009A_H, RSCANnGAFLECTRHH: <RSCFDn_base> + 009B_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|--------|----|----|----|------------|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | AFLDAE | — | — | — | AFLPN[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 17.29 RSCANnGAFLECTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 9 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 8 | AFLDAE | Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled. |
| 7 to 5 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 4 to 0 | AFLPN[4:0] | Receive Rule Table Page Number Configuration A page number can be selected in the range of page 0 (0000 _B) to page 11 (01011 _B). |

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000_B to 01011_B.

17.3.5.2 RSCANnGAFLCFG0 — Receive Rule Configuration Register 0

Access: RSCANnGAFLCFG0 register can be read/written in 32-bit units
 RSCANnGAFLCFG0L, RSCANnGAFLCFG0H registers can be read/written in 16-bit units
 RSCANnGAFLCFG0LL, RSCANnGAFLCFG0LH, RSCANnGAFLCFG0HL, RSCANnGAFLCFG0HH registers can be read/written in 8-bit units

Address: RSCANnGAFLCFG0: <RSCFDn_base> + 009C_H
 RSCANnGAFLCFG0L: <RSCFDn_base> + 009C_H, RSCANnGAFLCFG0H: <RSCFDn_base> + 009E_H
 RSCANnGAFLCFG0LL: <RSCFDn_base> + 009C_H, RSCANnGAFLCFG0LH: <RSCFDn_base> + 009D_H,
 RSCANnGAFLCFG0HL: <RSCFDn_base> + 009E_H, RSCANnGAFLCFG0HH: <RSCFDn_base> + 009F_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----------|-----|-----|-----|-----|-----|-----|-----|
| | RNC0[7:0] | | | | | | | | RNC1[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RNC2[7:0] | | | | | | | | — | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

Table 17.30 RSCANnGAFLCFG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 24 | RNC0[7:0] | Number of Rules for Channel 0 Set the number of receive rules used for channel 0. |
| 23 to 16 | RNC1[7:0] | Number of Rules for Channel 1 Set the number of receive rules used for channel 1. |
| 15 to 8 | RNC2[7:0] | Number of Rules for Channel 2 Set the number of receive rules used for channel 2. |
| 7 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Modify the RSCANnGAFLCFG0 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total number of rules allocated to each channel is not larger than the number of rules that can be registered for the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 2 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

17.3.5.3 RSCANnGAFLIDj — Receive Rule ID Register j (j = 0 to 15)

Access: RSCANnGAFLIDj register can be read/written in 32-bit units
RSCANnGAFLIDjL, RSCANnGAFLIDjH registers can be read/written in 16-bit units
RSCANnGAFLIDjLL, RSCANnGAFLIDjLH, RSCANnGAFLIDjHL, RSCANnGAFLIDjHH registers can be read/written in 8-bit units

Address: RSCANnGAFLIDj: <RSCFDn_base> + 0500_H + (10_H × j)
RSCANnGAFLIDjL: <RSCFDn_base> + 0500_H + (10_H × j),
RSCANnGAFLIDjH: <RSCFDn_base> + 0502_H + (10_H × j)
RSCANnGAFLIDjLL: <RSCFDn_base> + 0500_H + (10_H × j),
RSCANnGAFLIDjLH: <RSCFDn_base> + 0501_H + (10_H × j),
RSCANnGAFLIDjHL: <RSCFDn_base> + 0502_H + (10_H × j),
RSCANnGAFLIDjHH: <RSCFDn_base> + 0503_H + (10_H × j)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|-------------|-------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GAFLID E | GAFLR TR | GAFLB | GAFLID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.31 RSCANnGAFLIDj Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 | GAFLIDE | IDE Select 0: Standard ID 1: Extended ID |
| 30 | GAFLRTR | RTR Select 0: Data frame 1: Remote frame |
| 29 | GAFLLB | Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received |
| 28 to 0 | GAFLID[28:0] | ID setting Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0. |

Modify the RSCANnGAFLIDj register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

17.3.5.4 RSCANnGAFLMj — Receive Rule Mask Register j (j = 0 to 15)

Access: RSCANnGAFLMj register can be read/written in 32-bit units
 RSCANnGAFLMjL, RSCANnGAFLMjH registers can be read/written in 16-bit units
 RSCANnGAFLMjLL, RSCANnGAFLMjLH, RSCANnGAFLMjHL, RSCANnGAFLMjHH registers can be read/
 written in 8-bit units

Address: RSCANnGAFLMj: <RSCFDn_base> + 0504_H + (10_H × j)
 RSCANnGAFLMjL: <RSCFDn_base> + 0504_H + (10_H × j),
 RSCANnGAFLMjH: <RSCFDn_base> + 0506_H + (10_H × j)
 RSCANnGAFLMjLL: <RSCFDn_base> + 0504_H + (10_H × j),
 RSCANnGAFLMjLH: <RSCFDn_base> + 0505_H + (10_H × j),
 RSCANnGAFLMjHL: <RSCFDn_base> + 0506_H + (10_H × j),
 RSCANnGAFLMjHH: <RSCFDn_base> + 0507_H + (10_H × j)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|--------------|-----|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | GAFLID EM | GAFLR TRM | — | GAFLIDM[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLIDM[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.32 RSCANnGAFLMj Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31 | GAFLIDEM | IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared. |
| 30 | GAFLRTRM | RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared. |
| 29 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 28 to 0 | GAFLIDM[28:0] | ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared. |

Modify the RSCANnGAFLMj register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCANnGAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set all the GAFLIDM[28:0] bits to 0s at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

17.3.5.5 RSCANnGAFLP0_j — Receive Rule Pointer 0 Register j (j = 0 to 15)

Access: RSCANnGAFLP0_j register can be read/written in 32-bit units
 RSCANnGAFLP0_jL, RSCANnGAFLP0_jH registers can be read/written in 16-bit units
 RSCANnGAFLP0_jLL, RSCANnGAFLP0_jLH, RSCANnGAFLP0_jHL, RSCANnGAFLP0_jHH registers can be read/written in 8-bit units

Address: RSCANnGAFLP0_j: <RSCFDn_base> + 0508_H + (10_H × j)
 RSCANnGAFLP0_jL: <RSCFDn_base> + 0508_H + (10_H × j),
 RSCANnGAFLP0_jH: <RSCFDn_base> + 050A_H + (10_H × j)
 RSCANnGAFLP0_jLL: <RSCFDn_base> + 0508_H + (10_H × j),
 RSCANnGAFLP0_jLH: <RSCFDn_base> + 0509_H + (10_H × j),
 RSCANnGAFLP0_jHL: <RSCFDn_base> + 050A_H + (10_H × j),
 RSCANnGAFLP0_jHH: <RSCFDn_base> + 050B_H + (10_H × j)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|---------------|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GAFLDLC[3:0] | | | | GAFLPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLR MV | GAFLRMDP[6:0] | | | | | | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

Table 17.33 RSCANnGAFLP0_j Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31 to 28 | GAFLDLC[3:0] | Receive Rule DLC b31 b30 b29 b28 0 0 0 0: DLC check is disabled. 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes |
| 27 to 16 | GAFLPTR[11:0] | Receive Rule Label Set the 12-bit label information. |
| 15 | GAFLRMV | Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used. |
| 14 to 8 | GAFLRMDP[6:0] | Receive Buffer Number Select Set the receive buffer number to store receive messages. |
| 7 to 0 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |

Modify the RSCANnGAFLP0_j register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCANnRMNB register.

17.3.5.6 RSCANnGAFLP1_j — Receive Rule Pointer 1 Register j (j = 0 to 15)

Access: RSCANnGAFLP1_j register can be read/written in 32-bit units
 RSCANnGAFLP1_jL, RSCANnGAFLP1_jH registers can be read/written in 16-bit units
 RSCANnGAFLP1_jLL, RSCANnGAFLP1_jLH, RSCANnGAFLP1_jHL, RSCANnGAFLP1_jHH registers can be read/written in 8-bit units

Address: RSCANnGAFLP1_j: <RSCFDn_base> + 050C_H + (10_H × j)
 RSCANnGAFLP1_jL: <RSCFDn_base> + 050C_H + (10_H × j),
 RSCANnGAFLP1_jH: <RSCFDn_base> + 050E_H + (10_H × j)
 RSCANnGAFLP1_jLL: <RSCFDn_base> + 050C_H + (10_H × j),
 RSCANnGAFLP1_jLH: <RSCFDn_base> + 050D_H + (10_H × j),
 RSCANnGAFLP1_jHL: <RSCFDn_base> + 050E_H + (10_H × j),
 RSCANnGAFLP1_jHH: <RSCFDn_base> + 050F_H + (10_H × j)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | GAFLF DP16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLFDP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.34 RSCANnGAFLP1_j Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 31 to 17 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 16 to 8 | GAFLFDP[16:8] | Transmit/Receive FIFO Buffer k Select (Bit position – 8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected. |
| 7 to 0 | GAFLFDP[7:0] | Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected. |

Modify the RSCANnGAFLP1_j register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP[16:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers can be selected. However, when the GAFLRMV bit in the RSCANnGAFLP0_j register is set to 1 (messages are stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCANnCFCK register are set to 00_B (receive mode) or 10_B (gateway mode) are selectable.

17.3.6 Details of Receive Buffer-Related Registers

17.3.6.1 RSCANnRMNB — Receive Buffer Number Register

Access: RSCANnRMNB register can be read/written in 32-bit units
 RSCANnRMNBL, RSCANnRMNBH registers can be read/written in 16-bit units
 RSCANnRMNBLL, RSCANnRMNBHL, RSCANnRMNBHL, RSCANnRMNBHL registers can be read/written in 8-bit units

Address: RSCANnRMNB: <RSCFDn_base> + 00A4_H
 RSCANnRMNBL: <RSCFDn_base> + 00A4_H, RSCANnRMNBH: <RSCFDn_base> + 00A6_H
 RSCANnRMNBLL: <RSCFDn_base> + 00A4_H, RSCANnRMNBHL: <RSCFDn_base> + 00A5_H,
 RSCANnRMNBHL: <RSCFDn_base> + 00A6_H, RSCANnRMNBHL: <RSCFDn_base> + 00A7_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | NRXMB[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.35 RSCANnRMNB Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 8 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 7 to 0 | NRXMB[7:0] | Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 48. |

Modify the RSCANnRMNB register only in global reset mode.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CANFD module. The maximum value is $16 \times$ (number of channels).

Setting all of these bits to 0 makes receive buffers unavailable.

17.3.6.2 RSCANnRMNDy — Receive Buffer New Data Register y (y = 0, 1)

Access: RSCANnRMNDy register can be read/written in 32-bit units
 RSCANnRMNDyL, RSCANnRMNDyH registers can be read/written in 16-bit units
 RSCANnRMNDyLL, RSCANnRMNDyLH, RSCANnRMNDyHL, RSCANnRMNDyHH registers can be read/written in 8-bit units

Address: RSCANnRMNDy: <RSCFDn_base> + 00A8_H + (04_H × y)
 RSCANnRMNDyL: <RSCFDn_base> + 00A8_H + (04_H × y),
 RSCANnRMNDyH: <RSCFDn_base> + 00AA_H + (04_H × y)
 RSCANnRMNDyLL: <RSCFDn_base> + 00A8_H + (04_H × y),
 RSCANnRMNDyLH: <RSCFDn_base> + 00A9_H + (04_H × y),
 RSCANnRMNDyHL: <RSCFDn_base> + 00AA_H + (04_H × y),
 RSCANnRMNDyHH: <RSCFDn_base> + 00AB_H + (04_H × y)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.36 RSCANnRMNDy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | RMNSq | Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q. When y = 1, these bits are reserved. When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | RMNSq | Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q. |

Write 0 to the RSCANnRMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 47)

Each RMNSq flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. It takes ten pclk clock cycles to store a message.

These flags are cleared to 0 in global reset mode.

17.3.6.3 RSCANnRMIDq — Receive Buffer ID Register q (q = 0 to 47)

Access: RSCANnRMIDq register can be read only in 32-bit units
 RSCANnRMIDqL, RSCANnRMIDqH registers can be read only in 16-bit units
 RSCANnRMIDqLL, RSCANnRMIDqLH, RSCANnRMIDqHL, RSCANnRMIDqHH registers can be read only in 8-bit units

Address: RSCANnRMIDq: $\langle \text{RSCFDn_base} \rangle + 0600_{\text{H}} + (10_{\text{H}} \times q)$
 RSCANnRMIDqL: $\langle \text{RSCFDn_base} \rangle + 0600_{\text{H}} + (10_{\text{H}} \times q)$,
 RSCANnRMIDqH: $\langle \text{RSCFDn_base} \rangle + 0602_{\text{H}} + (10_{\text{H}} \times q)$
 RSCANnRMIDqLL: $\langle \text{RSCFDn_base} \rangle + 0600_{\text{H}} + (10_{\text{H}} \times q)$,
 RSCANnRMIDqLH: $\langle \text{RSCFDn_base} \rangle + 0601_{\text{H}} + (10_{\text{H}} \times q)$,
 RSCANnRMIDqHL: $\langle \text{RSCFDn_base} \rangle + 0602_{\text{H}} + (10_{\text{H}} \times q)$,
 RSCANnRMIDqHH: $\langle \text{RSCFDn_base} \rangle + 0603_{\text{H}} + (10_{\text{H}} \times q)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-------|----|-------------|----|----|----|----|----|----|----|----|----|----|----|----|
| | RMIDE | RMRTR | — | RMID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.37 RSCANnRMIDq Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 | RMIDE | Receive Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | RMRTR | Receive Buffer RTR 0: Data frame 1: Remote frame |
| 29 | Reserved | These bits are read as the value after reset. |
| 28 to 0 | RMID[28:0] | Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0. |

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

17.3.6.4 RSCANnRMPTRq — Receive Buffer Pointer Register q (q = 0 to 47)

Access: RSCANnRMPTRq register can be read only in 32-bit units
 RSCANnRMPTRqL, RSCANnRMPTRqH registers can be read only in 16-bit units
 RSCANnRMPTRqLL, RSCANnRMPTRqLH, RSCANnRMPTRqHL, RSCANnRMPTRqHH registers can be read only in 8-bit units

Address: RSCANnRMPTRq: <RSCFDn_base> + 0604_H + (10_H × q)
 RSCANnRMPTRqL: <RSCFDn_base> + 0604_H + (10_H × q),
 RSCANnRMPTRqH: <RSCFDn_base> + 0606_H + (10_H × q)
 RSCANnRMPTRqLL: <RSCFDn_base> + 0604_H + (10_H × q),
 RSCANnRMPTRqLH: <RSCFDn_base> + 0605_H + (10_H × q),
 RSCANnRMPTRqHL: <RSCFDn_base> + 0606_H + (10_H × q),
 RSCANnRMPTRqHH: <RSCFDn_base> + 0607_H + (10_H × q)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|----|----|----|-------------|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RMDLC[3:0] | | | | RMPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.38 RSCANnRMPTRq Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 28 | RMDLC[3:0] | Receive Buffer DLC Data b31 b30 b29 b28 0 0 0 0: No data byte 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes |
| 27 to 16 | RMPTR[11:0] | Receive Buffer Label Data Label information of the received message. |
| 15 to 0 | RMTS[15:0] | Receive Buffer Timestamp Data Timestamp value of the received message. |

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

17.3.6.5 RSCANnRMDF0_q — Receive Buffer Data Field 0 Register q (q = 0 to 47)

Access: RSCANnRMDF0_q register can be read only in 32-bit units
 RSCANnRMDF0_qL, RSCANnRMDF0_qH registers can be read only in 16-bit units
 RSCANnRMDF0_qLL, RSCANnRMDF0_qLH, RSCANnRMDF0_qHL, RSCANnRMDF0_qHH registers can be read only in 8-bit units

Address: RSCANnRMDF0_q: <RSCFDn_base> + 0608_H + (10_H × q)
 RSCANnRMDF0_qL: <RSCFDn_base> + 0608_H + (10_H × q),
 RSCANnRMDF0_qH: <RSCFDn_base> + 060A_H + (10_H × q)
 RSCANnRMDF0_qLL: <RSCFDn_base> + 0608_H + (10_H × q),
 RSCANnRMDF0_qLH: <RSCFDn_base> + 0609_H + (10_H × q),
 RSCANnRMDF0_qHL: <RSCFDn_base> + 060A_H + (10_H × q),
 RSCANnRMDF0_qHH: <RSCFDn_base> + 060B_H + (10_H × q)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| | RMDB3[7:0] | | | | | | | | RMDB2[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMDB1[7:0] | | | | | | | | RMDB0[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.39 RSCANnRMDF0_q Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 24 | RMDB3[7:0] | Receive Buffer Data Byte 3 |
| 23 to 16 | RMDB2[7:0] | Receive Buffer Data Byte 2 |
| 15 to 8 | RMDB1[7:0] | Receive Buffer Data Byte 1 |
| 7 to 0 | RMDB0[7:0] | Receive Buffer Data Byte 0 |
| | | Data for a message stored in the receive buffer can be read. |

When the RMDLC[3:0] value in the RSCANnRMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

17.3.6.6 RSCANnRMDF1_q — Receive Buffer Data Field 1 Register q (q = 0 to 47)

Access: RSCANnRMDF1_q register can be read only in 32-bit units
 RSCANnRMDF1_qL, RSCANnRMDF1_qH register can be read only in 16-bit units
 RSCANnRMDF1_qLL, RSCANnRMDF1_qLH, RSCANnRMDF1_qHL, RSCANnRMDF1_qHH registers can be read only in 8-bit units

Address: RSCANnRMDF1_q: <RSCFDn_base> + 060C_H + (10_H × q)
 RSCANnRMDF1_qL: <RSCFDn_base> + 060C_H + (10_H × q),
 RSCANnRMDF1_qH: <RSCFDn_base> + 060E_H + (10_H × q)
 RSCANnRMDF1_qLL: <RSCFDn_base> + 060C_H + (10_H × q),
 RSCANnRMDF1_qLH: <RSCFDn_base> + 060D_H + (10_H × q),
 RSCANnRMDF1_qHL: <RSCFDn_base> + 060E_H + (10_H × q),
 RSCANnRMDF1_qHH: <RSCFDn_base> + 060F_H + (10_H × q)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| | RMDB7[7:0] | | | | | | | | RMDB6[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMDB5[7:0] | | | | | | | | RMDB4[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.40 RSCANnRMDF1_q Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 24 | RMDB7[7:0] | Receive Buffer Data Byte 7 |
| 23 to 16 | RMDB6[7:0] | Receive Buffer Data Byte 6 |
| 15 to 8 | RMDB5[7:0] | Receive Buffer Data Byte 5 |
| 7 to 0 | RMDB4[7:0] | Data for a message stored in the receive buffer can be read. |

When the RMDLC[3:0] value in the RSCANnRMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

17.3.7 Details of Receive FIFO Buffer-Related Registers

17.3.7.1 RSCANnRFCCx — Receive FIFO Buffer Configuration and Control Register x (x = 0 to 7)

Access: RSCANnRFCCx register can be read/written in 32-bit units
 RSCANnRFCCxL, RSCANnRFCCxH registers can be read/written in 16-bit units
 RSCANnRFCCxLL, RSCANnRFCCxLH, RSCANnRFCCxHL, RSCANnRFCCxHH registers can be read/written in 8-bit units

Address: RSCANnRFCCx: <RSCFDn_base> + 00B8_H + (04_H × x)
 RSCANnRFCCxL: <RSCFDn_base> + 00B8_H + (04_H × x),
 RSCANnRFCCxH: <RSCFDn_base> + 00BA_H + (04_H × x)
 RSCANnRFCCxLL: <RSCFDn_base> + 00B8_H + (04_H × x),
 RSCANnRFCCxLH: <RSCFDn_base> + 00B9_H + (04_H × x),
 RSCANnRFCCxHL: <RSCFDn_base> + 00BA_H + (04_H × x),
 RSCANnRFCCxHH: <RSCFDn_base> + 00BB_H + (04_H × x)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|-----|------|----|-----------|-----|-----|----|----|----|----|----|----|------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFIGCV[2:0] | | | RFIM | — | RFDC[2:0] | | | — | — | — | — | — | — | RFIE | RFE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R | R | R | R | R | R/W | R/W |

Table 17.41 RSCANnRFCCx Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 16 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 15 to 13 | RFIGCV[2:0] | Receive FIFO Interrupt Request Timing Select $b_{15} \ b_{14} \ b_{13}$ 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full. |
| 12 | RFIM | Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received. |
| 11 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 10 to 8 | RFDC[2:0] | Receive FIFO Buffer Depth Configuration $b_{10} \ b_9 \ b_8$ 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages |

Table 17.41 RSCANnRFCCx Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 2 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 1 | RFIE | Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled. |
| 0 | RFE | Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used. |

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit is set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 enables the use of receive FIFO buffers. Clearing this bit to 0 sets the RFEMP flag in the RSCANnRFSTs_x register to 1 (buffer empty). Modify this bit in global operating mode or global test mode.

Set this bit to 1 with another instruction after the settings to all bits in the RSCANnRFCC_x register have been done.

This bit is cleared to 0 in global reset mode.

17.3.7.2 RSCANnRFSTStx — Receive FIFO Buffer Status Register x (x = 0 to 7)

Access: RSCANnRFSTStx register can be read/written in 32-bit units
 RSCANnRFSTStxL, RSCANnRFSTStxH registers can be read/written in 16-bit units
 RSCANnRFSTStxLL, RSCANnRFSTStxLH, RSCANnRFSTStxHL, RSCANnRFSTStxHH registers can be read/written in 8-bit units

Address: RSCANnRFSTStx: <RSCFDn_base> + 00D8_H + (04_H × x)
 RSCANnRFSTStxL: <RSCFDn_base> + 00D8_H + (04_H × x),
 RSCANnRFSTStxH: <RSCFDn_base> + 00DA_H + (04_H × x)
 RSCANnRFSTStxLL: <RSCFDn_base> + 00D8_H + (04_H × x),
 RSCANnRFSTStxLH: <RSCFDn_base> + 00D9_H + (04_H × x),
 RSCANnRFSTStxHL: <RSCFDn_base> + 00DA_H + (04_H × x),
 RSCANnRFSTStxHH: <RSCFDn_base> + 00DB_H + (04_H × x)

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|------|-------|-------|-------|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFMC[7:0] | | | | | | | — | — | — | — | RFIF | RFMLT | RFFLL | RFEMP | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.42 RSCANnRFSTStx Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 15 to 8 | RFMC[7:0] | Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed. |
| 7 to 4 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 3 | RFIF | Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present. |
| 2 | RFMLT | Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost. |
| 1 | RFFLL | Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full. |
| 0 | RFEMP | Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty). |

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RSCANnRFCCx register is set to 0.

This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCANnRFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCANnRFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCANnRFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCANnRFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message is stored in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

17.3.7.3 RSCANnRFPCTR_x — Receive FIFO Buffer Pointer Control Register x (x = 0 to 7)

Access: RSCANnRFPCTR_x register can only be written in 32-bit units
 RSCANnRFPCTR_{xL}, RSCANnRFPCTR_{xH} registers can only be written in 16-bit units
 RSCANnRFPCTR_{xLL}, RSCANnRFPCTR_{xLH}, RSCANnRFPCTR_{xHL}, RSCANnRFPCTR_{xHH} registers can only be written in 8-bit units

Address: RSCANnRFPCTR_x: <RSCFDn_base> + 00F8_H + (04_H × x)
 RSCANnRFPCTR_{xL}: <RSCFDn_base> + 00F8_H + (04_H × x),
 RSCANnRFPCTR_{xH}: <RSCFDn_base> + 00FA_H + (04_H × x)
 RSCANnRFPCTR_{xLL}: <RSCFDn_base> + 00F8_H + (04_H × x),
 RSCANnRFPCTR_{xLH}: <RSCFDn_base> + 00F9_H + (04_H × x),
 RSCANnRFPCTR_{xHL}: <RSCFDn_base> + 00FA_H + (04_H × x),
 RSCANnRFPCTR_{xHH}: <RSCFDn_base> + 00FB_H + (04_H × x)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | RFPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.43 RSCANnRFPCTR_x Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 8 | Reserved | The write value should be the value after reset. |
| 7 to 0 | RFPC[7:0] | Receive FIFO Pointer Control When these bits are set to FF _H , the read pointer moves to the next unread message in the receive FIFO buffer. |

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCANnRFST_{Sx} register is decremented by 1. Read the RSCANnRFID_x, RSCANnRFPTR_x, RSCANnRFDF0_x, and RSCANnRFDF1_x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RSCANnRFCC_x register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCANnRFST_{Sx} register is 0 (the receive FIFO buffer contains unread messages).

17.3.7.4 RSCANnRFIDx — Receive FIFO Buffer Access ID Register x (x = 0 to 7)

Access: RSCANnRFIDx register can be read only in 32-bit units
 RSCANnRFIDxL, RSCANnRFIDxH registers can be read only in 16-bit units
 RSCANnRFIDxLL, RSCANnRFIDxLH, RSCANnRFIDxHL, RSCANnRFIDxHH registers can be read only in 8-bit units

Address: RSCANnRFIDx: <RSCFDn_base> + 0E00_H + (10_H × x)
 RSCANnRFIDxL: <RSCFDn_base> + 0E00_H + (10_H × x),
 RSCANnRFIDxH: <RSCFDn_base> + 0E02_H + (10_H × x)
 RSCANnRFIDxLL: <RSCFDn_base> + 0E00_H + (10_H × x),
 RSCANnRFIDxLH: <RSCFDn_base> + 0E01_H + (10_H × x),
 RSCANnRFIDxHL: <RSCFDn_base> + 0E02_H + (10_H × x),
 RSCANnRFIDxHH: <RSCFDn_base> + 0E03_H + (10_H × x)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-------|----|-------------|----|----|----|----|----|----|----|----|----|----|----|----|
| | RFIDE | RFRTR | — | RFID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.44 RSCANnRFIDx Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 | RFIDE | Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | RFRTR | Receive FIFO Buffer RTR 0: Data frame 1: Remote frame |
| 29 | Reserved | These bits are read as the value after reset. |
| 28 to 0 | RFID[28:0] | Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0. |

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

17.3.7.5 RSCANnRFPTRx — Receive FIFO Buffer Access Pointer Register x (x = 0 to 7)

Access: RSCANnRFPTRx register can be read only in 32-bit units
 RSCANnRFPTRxL, RSCANnRFPTRxH registers can be read only in 16-bit units
 RSCANnRFPTRxLL, RSCANnRFPTRxLH, RSCANnRFPTRxHL, RSCANnRFPTRxHH registers can be read only in 8-bit units

Address: RSCANnRFPTRx: <RSCFDn_base> + 0E04_H + (10_H × x)
 RSCANnRFPTRxL: <RSCFDn_base> + 0E04_H + (10_H × x),
 RSCANnRFPTRxH: <RSCFDn_base> + 0E06_H + (10_H × x)
 RSCANnRFPTRxLL: <RSCFDn_base> + 0E04_H + (10_H × x),
 RSCANnRFPTRxLH: <RSCFDn_base> + 0E05_H + (10_H × x),
 RSCANnRFPTRxHL: <RSCFDn_base> + 0E06_H + (10_H × x),
 RSCANnRFPTRxHH: <RSCFDn_base> + 0E07_H + (10_H × x)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|-------------|----|----|----|----|----|----|----|----|----|----|----|
| | RFDLC[3:0] | | | | RFPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.45 RSCANnRFPTRx Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 28 | RFDLC[3:0] | Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes |
| 27 to 16 | RFPTR[11:0] | Receive FIFO Buffer Label Data Label information of the received message can be read. |
| 15 to 0 | RFTS[15:0] | Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read. |

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

17.3.7.6 RSCANnRFDF0_x — Receive FIFO Buffer Access Data Field 0 Register x (x = 0 to 7)

Access: RSCANnRFDF0_x register can be read-only in 32-bit units
RSCANnRFDF0_xL, RSCANnRFDF0_xH registers can be read only in 16-bit units
RSCANnRFDF0_xLL, RSCANnRFDF0_xLH, RSCANnRFDF0_xHL, RSCANnRFDF0_xHH registers can be read only in 8-bit units

Address: RSCANnRFDF0_x: <RSCFDn_base> + 0E08_H + (10_H × x)
RSCANnRFDF0_xL: <RSCFDn_base> + 0E08_H + (10_H × x),
RSCANnRFDF0_xH: <RSCFDn_base> + 0E0A_H + (10_H × x)
RSCANnRFDF0_xLL: <RSCFDn_base> + 0E08_H + (10_H × x),
RSCANnRFDF0_xLH: <RSCFDn_base> + 0E09_H + (10_H × x),
RSCANnRFDF0_xHL: <RSCFDn_base> + 0E0A_H + (10_H × x),
RSCANnRFDF0_xHH: <RSCFDn_base> + 0E0B_H + (10_H × x)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| | RFDB3[7:0] | | | | | | | | RFDB2[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFDB1[7:0] | | | | | | | | RFDB0[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.46 RSCANnRFDF0_x Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 24 | RFDB3[7:0] | Receive FIFO Buffer Data Byte 3 |
| 23 to 16 | RFDB2[7:0] | Receive FIFO Buffer Data Byte 2 |
| 15 to 8 | RFDB1[7:0] | Receive FIFO Buffer Data Byte 1 |
| 7 to 0 | RFDB0[7:0] | Data for a message stored in the receive FIFO buffer can be read. |

When the RFDLC[3:0] value in the RSCANnRFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

17.3.7.7 RSCANnRFDF1_x — Receive FIFO Buffer Access Data Field 1 Register x (x = 0 to 7)

Access: RSCANnRFDF1_x register can be read only in 32-bit units
RSCANnRFDF1_xL, RSCANnRFDF1_xH registers can be read only in 16-bit units
RSCANnRFDF1_xLL, RSCANnRFDF1_xLH, RSCANnRFDF1_xHL, RSCANnRFDF1_xHH registers can be read only in 8-bit units

Address: RSCANnRFDF1_x: <RSCFDn_base> + 0E0C_H + (10_H × x)
RSCANnRFDF1_xL: <RSCFDn_base> + 0E0C_H + (10_H × x),
RSCANnRFDF1_xH: <RSCFDn_base> + 0E0E_H + (10_H × x)
RSCANnRFDF1_xLL: <RSCFDn_base> + 0E0C_H + (10_H × x),
RSCANnRFDF1_xLH: <RSCFDn_base> + 0E0D_H + (10_H × x),
RSCANnRFDF1_xHL: <RSCFDn_base> + 0E0E_H + (10_H × x),
RSCANnRFDF1_xHH: <RSCFDn_base> + 0E0F_H + (10_H × x)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RFDB7[7:0] | | | | | | | | RFDB6[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFDB5[7:0] | | | | | | | | RFDB4[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.47 RSCANnRFDF1_x Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 24 | RFDB7[7:0] | Receive FIFO Buffer Data Byte 7 |
| 23 to 16 | RFDB6[7:0] | Receive FIFO Buffer Data Byte 6 |
| 15 to 8 | RFDB5[7:0] | Receive FIFO Buffer Data Byte 5 |
| 7 to 0 | RFDB4[7:0] | Data for a message stored in the receive FIFO buffer can be read. |

When the RFDLC[3:0] value in the RSCANnRFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

17.3.8 Details of Transmit/Receive FIFO Buffer-Related Registers

17.3.8.1 RSCANnCFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 8)

Access: RSCANnCFCCk register can be read/written in 32-bit units
 RSCANnCFCCkL, RSCANnCFCCkH registers can be read/written in 16-bit units
 RSCANnCFCCkLL, RSCANnCFCCkLH, RSCANnCFCCkHL, RSCANnCFCCkHH registers can be read/written in 8-bit units

Address: RSCANnCFCCk: <RSCFDn_base> + 0118_H + (04_H × k)
 RSCANnCFCCkL: <RSCFDn_base> + 0118_H + (04_H × k),
 RSCANnCFCCkH: <RSCFDn_base> + 011A_H + (04_H × k)
 RSCANnCFCCkLL: <RSCFDn_base> + 0118_H + (04_H × k),
 RSCANnCFCCkLH: <RSCFDn_base> + 0119_H + (04_H × k),
 RSCANnCFCCkHL: <RSCFDn_base> + 011A_H + (04_H × k),
 RSCANnCFCCkHH: <RSCFDn_base> + 011B_H + (04_H × k)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|-----|-----|------|-----|-----------|-----|------------|-----|-----|-----|-------|--------|----------|--------|-----|
| | CFITT[7:0] | | | | | | | CFTML[3:0] | | | | CFITR | CFITSS | CFM[1:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFIGCV[2:0] | | | CFIM | — | CFDC[2:0] | | | — | — | — | — | — | CFTXIE | CFRXIE | CFE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 17.48 RSCANnCFCCk Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 24 | CFITT[7:0] | Set a message transmission interval. Set Value: 00 _H to FF _H |
| 23 to 20 | CFTML[3:0] | Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer. |
| 19 | CFITR | Transmit/Receive FIFO Interval Timer Resolution 0: Clock obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits 1: Clock obtained by dividing pclk/2 by “the value of ITRCP[15:0] bits × 10” |
| 18 | CFITSS | Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the bit time clock for the channel to which the FIFO is linked. |
| 17, 16 | CFM[1:0] | Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited |
| 15 to 13 | CFIGCV[2:0] | Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full. |

Table 17.48 RSCANnCFCK Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 12 | CFIM | Transmit/Receive FIFO Interrupt Source Select 0: <ul style="list-style-type: none"> Receive mode/gateway mode When the number of received messages meets the condition set by the CFICV[2:0] bits, a FIFO receive interrupt request is generated. Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: <ul style="list-style-type: none"> Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted. |
| 11 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 10 to 8 | CFDC[2:0] | Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages |
| 7 to 3 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 2 | CFTXIE | Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled. |
| 1 | CFRXIE | Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled. |
| 0 | CFE | Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used. |

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as $m = k/3$ (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be $((16 \times m) + CFTML[3:0])$ (see **Table 17.15**).

See **Table 17.13** and **Table 17.14**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the same channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RSCANnGCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RSCANnGCFG register × 10).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the bit time clock of the channel to which the FIFO is linked is the count source of the interval timer.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CCTXIE Bit

When this bit is set to 1 and the CCTXIF flag in the RSCANnCFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit when the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCANnCFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit when the CFE bit is set to 0.

CFE Bit

Setting this bit to 1 enables the transmit/receive FIFO buffers.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCANnCFCCk register have been set, set this bit to 1 by using another instruction.

17.3.8.2 RSCANnCFSTSk — Transmit/receive FIFO Buffer Status Register k (k = 0 to 8)

Access: RSCANnCFSTSk register can be read/written in 32-bit units
 RSCANnCFSTSkL, RSCANnCFSTSkH registers can be read/written in 16-bit units
 RSCANnCFSTSkLL, RSCANnCFSTSkLH, RSCANnCFSTSkHL, RSCANnCFSTSkHH registers can be read/written in 8-bit units

Address: RSCANnCFSTSk: <RSCFDn_base> + 0178_H + (04_H × k)
 RSCANnCFSTSkL: <RSCFDn_base> + 0178_H + (04_H × k),
 RSCANnCFSTSkH: <RSCFDn_base> + 017A_H + (04_H × k)
 RSCANnCFSTSkLL: <RSCFDn_base> + 0178_H + (04_H × k),
 RSCANnCFSTSkLH: <RSCFDn_base> + 0179_H + (04_H × k),
 RSCANnCFSTSkHL: <RSCFDn_base> + 017A_H + (04_H × k),
 RSCANnCFSTSkHH: <RSCFDn_base> + 017B_H + (04_H × k)

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|--------|--------|-------|-------|-------|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFMC[7:0] | | | | | | | — | — | — | CFTXIF | CFRXIF | CFMLT | CFLL | CFEMP | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.49 RSCANnCFSTSk Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 15 to 8 | CFMC[7:0] | Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer. |
| 7 to 5 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 4 | CFTXIF | Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present. |
| 3 | CFRXIF | Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present. |
| 2 | CFMLT | Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost. |
| 1 | CFLL | Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full. |
| 0 | CFEMP | Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty). |

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values according to the setting of the CFM[1:0] bits in the RSCANnCFCCk register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode
- When the CFE bit in the RSCANnCFCCk register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the interrupt source selected by the CFIM bit in the RSCANnCFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the interrupt source selected by the CFIM bit in the RSCANnCFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFLL Flag

The CFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCANnCFCCk register.

The CFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCANnCFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RSCANnCFPCTRk register after data was written to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers.

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write "0" to the given flag and "1" to other flags.

17.3.8.3 RSCANnCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register k (k = 0 to 8)

Access: RSCANnCFPCTRk register can only be written in 32-bit units
 RSCANnCFPCTRkL, RSCANnCFPCTRkH registers can only be written in 16-bit units
 RSCANnCFPCTRkLL, RSCANnCFPCTRkLH, RSCANnCFPCTRkHL, RSCANnCFPCTRkHH registers can only be written in 8-bit units

Address: RSCANnCFPCTRk: <RSCFDn_base> + 01D8_H + (04_H × k)
 RSCANnCFPCTRkL: <RSCFDn_base> + 01D8_H + (04_H × k),
 RSCANnCFPCTRkH: <RSCFDn_base> + 01DA_H + (04_H × k)
 RSCANnCFPCTRkLL: <RSCFDn_base> + 01D8_H + (04_H × k),
 RSCANnCFPCTRkLH: <RSCFDn_base> + 01D9_H + (04_H × k),
 RSCANnCFPCTRkHL: <RSCFDn_base> + 01DA_H + (04_H × k),
 RSCANnCFPCTRkHH: <RSCFDn_base> + 01DB_H + (04_H × k)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | CFPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.50 RSCANnCFPCTRk Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 8 | Reserved | The write value should be the value after reset. |
| 7 to 0 | CFPC[7:0] | Transmit/Receive FIFO Pointer Control <ul style="list-style-type: none"> Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. Gateway mode: Setting prohibited |

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCANnCFCCk register is 00_B):
 Writing FF_H to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCANnCFSTSk register is decremented by 1. Read the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.
 When writing FF_H to these bits, make sure that the CFE bit in the RSCANnCFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCANnCFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).

- Transmit mode (CFM[1:0] value in the RSCANnCFCCk register is 01_B):
Writing FF_H to the CFPC[7:0] bits stores the data written to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1. Write transmit messages to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers before writing FF_H to the CFPC[7:0] bits.
When writing FF_H to these bits, make sure that the CFE bit in the RSCANnCFCCk register is set to 1 and the CFFLL flag in the RSCANnCFSTSk register is 0 (the transmit/receive FIFO buffer is not full).
- Gateway mode (CFM[1:0] value in the RSCANnCFCCk register is 10_B):
Setting prohibited

17.3.8.4 RSCANnCFIDk — Transmit/receive FIFO Buffer Access ID Register k (k = 0 to 8)

Access: RSCANnCFIDk register can be read/written in 32-bit units
 RSCANnCFIDkL, RSCANnCFIDkH registers can be read/written in 16-bit units
 RSCANnCFIDkLL, RSCANnCFIDkLH, RSCANnCFIDkHL, RSCANnCFIDkHH registers can be read/written in 8-bit units

Address: RSCANnCFIDk: <RSCFDn_base> + 0E80_H + (10_H × k)
 RSCANnCFIDkL: <RSCFDn_base> + 0E80_H + (10_H × k),
 RSCANnCFIDkH: <RSCFDn_base> + 0E82_H + (10_H × k)
 RSCANnCFIDkLL: <RSCFDn_base> + 0E80_H + (10_H × k),
 RSCANnCFIDkLH: <RSCFDn_base> + 0E81_H + (10_H × k),
 RSCANnCFIDkHL: <RSCFDn_base> + 0E82_H + (10_H × k),
 RSCANnCFIDkHH: <RSCFDn_base> + 0E83_H + (10_H × k)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-------|-------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CFIDE | CFRTR | THLEN | CFID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.51 RSCANnCFIDk Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 | CFIDE | Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | CFRTR | Transmit/Receive FIFO Buffer RTR 0: Data frame 1: Remote frame |
| 29 | THLEN | Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01 _B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer. |
| 28 to 0 | CFID[28:0] | Transmit/Receive FIFO Buffer ID Data <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0. |

This register can be written only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

These bits set the ID of the message to be transmitted from the transmit/receive FIFO buffer when the CFM[1:0] value is 01_B.

17.3.8.5 RSCANnCFPTRk — Transmit/receive FIFO Buffer Access Pointer Register k (k = 0 to 8)

Access: RSCANnCFPTRk register can be read/written in 32-bit units
 RSCANnCFPTRkL, RSCANnCFPTRkH registers can be read/written in 16-bit units
 RSCANnCFPTRkLL, RSCANnCFPTRkLH, RSCANnCFPTRkHL, RSCANnCFPTRkHH registers can be read/written in 8-bit units

Address: RSCANnCFPTRk: <RSCFDn_base> + 0E84_H + (10_H × k)
 RSCANnCFPTRkL: <RSCFDn_base> + 0E84_H + (10_H × k),
 RSCANnCFPTRkH: <RSCFDn_base> + 0E86_H + (10_H × k)
 RSCANnCFPTRkLL: <RSCFDn_base> + 0E84_H + (10_H × k),
 RSCANnCFPTRkLH: <RSCFDn_base> + 0E85_H + (10_H × k),
 RSCANnCFPTRkHL: <RSCFDn_base> + 0E86_H + (10_H × k),
 RSCANnCFPTRkHH: <RSCFDn_base> + 0E87_H + (10_H × k)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CFDLC[3:0] | | | | CFPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.52 RSCANnCFPTRk Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 28 | CFDLC[3:0] | Transmit/Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes |
| 27 to 16 | CFPTR[11:0] | Transmit/Receive FIFO Buffer Label Data <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read. |
| 15 to 0 | CFTS[15:0] | Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is 00 _B (receive mode). The timestamp value of the received message can be read. |

This register is writable only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If the data length is set to 1001_B or more, the actual transmit data defaults to 8 bytes.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

17.3.8.6 RSCANnCFDF0_k — Transmit/receive FIFO Buffer Access Data Field 0 Register k (k = 0 to 8)

Access: RSCANnCFDF0_k register can be read/written in 32-bit units
 RSCANnCFDF0_kL, RSCANnCFDF0_kH registers can be read/written in 16-bit units
 RSCANnCFDF0_kLL, RSCANnCFDF0_kLH, RSCANnCFDF0_kHL, RSCANnCFDF0_kHH registers can be read/written in 8-bit units

Address: RSCANnCFDF0_k: <RSCFDn_base> + 0E88_H + (10_H × k)
 RSCANnCFDF0_kL: <RSCFDn_base> + 0E88_H + (10_H × k),
 RSCANnCFDF0_kH: <RSCFDn_base> + 0E8A_H + (10_H × k)
 RSCANnCFDF0_kLL: <RSCFDn_base> + 0E88_H + (10_H × k),
 RSCANnCFDF0_kLH: <RSCFDn_base> + 0E89_H + (10_H × k),
 RSCANnCFDF0_kHL: <RSCFDn_base> + 0E8A_H + (10_H × k),
 RSCANnCFDF0_kHH: <RSCFDn_base> + 0E8B_H + (10_H × k)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CFDB3[7:0] | | | | | | | | CFDB2[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFDB1[7:0] | | | | | | | | CFDB0[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.53 RSCANnCFDF0_k Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 24 | CFDB3[7:0] | Transmit/Receive FIFO Buffer Data Byte 3 |
| 23 to 16 | CFDB2[7:0] | Transmit/Receive FIFO Buffer Data Byte 2 |
| 15 to 8 | CFDB1[7:0] | Transmit/Receive FIFO Buffer Data Byte 1 |
| 7 to 0 | CFDB0[7:0] | <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read. |

This register can be written only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode).

This register can be read only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCANnCFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

17.3.8.7 RSCANnCFDF1_k — Transmit/receive FIFO Buffer Access Data Field 1 Register k (k = 0 to 8)

Access: RSCANnCFDF1_k register can be read/written in 32-bit units
 RSCANnCFDF1_kL, RSCANnCFDF1_kH registers can be read/written in 16-bit units
 RSCANnCFDF1_kLL, RSCANnCFDF1_kLH, RSCANnCFDF1_kHL, RSCANnCFDF1_kHH registers can be read/written in 8-bit units

Address: RSCANnCFDF1_k: <RSCFDn_base> + 0E8C_H + (10_H × k)
 RSCANnCFDF1_kL: <RSCFDn_base> + 0E8C_H + (10_H × k),
 RSCANnCFDF1_kH: <RSCFDn_base> + 0E8E_H + (10_H × k)
 RSCANnCFDF1_kLL: <RSCFDn_base> + 0E8C_H + (10_H × k),
 RSCANnCFDF1_kLH: <RSCFDn_base> + 0E8D_H + (10_H × k),
 RSCANnCFDF1_kHL: <RSCFDn_base> + 0E8E_H + (10_H × k),
 RSCANnCFDF1_kHH: <RSCFDn_base> + 0E8F_H + (10_H × k)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CFDB7[7:0] | | | | | | | | CFDB6[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFDB5[7:0] | | | | | | | | CFDB4[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.54 RSCANnCFDF1_k Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 24 | CFDB7[7:0] | Transmit/Receive FIFO Buffer Data Byte 7 |
| 23 to 16 | CFDB6[7:0] | Transmit/Receive FIFO Buffer Data Byte 6 |
| 15 to 8 | CFDB5[7:0] | Transmit/Receive FIFO Buffer Data Byte 5 |
| 7 to 0 | CFDB4[7:0] | Transmit/Receive FIFO Buffer Data Byte 4 <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read. |

This register is writable only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCANnCFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

17.3.9 Details of FIFO Status-Related Registers

17.3.9.1 RSCANnFESTS — FIFO Empty Status Register

Access: RSCANnFESTS registers can be read only in 32-bit units
 RSCANnFESTSL, RSCANnFESTSH registers can be read only in 16-bit units
 RSCANnFESTSLL, RSCANnFESTSLH, RSCANnFESTSHL, RSCANnFESTSHH registers can be read only in 8-bit units

Address: RSCANnFESTS: <RSCFDn_base> + 0238_H
 RSCANnFESTSL: <RSCFDn_base> + 0238_H, RSCANnFESTSH: <RSCFDn_base> + 023A_H
 RSCANnFESTSLL: <RSCFDn_base> + 0238_H, RSCANnFESTSLH: <RSCFDn_base> + 0239_H,
 RSCANnFESTSHL: <RSCFDn_base> + 023A_H, RSCANnFESTSHH: <RSCFDn_base> + 023B_H

Value after reset: 03FF FFFF_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CF8EMP |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | CF7EMP | CF6EMP | CF5EMP | CF4EMP | CF3EMP | CF2EMP | CF1EMP | CF0EMP | RF7EMP | RF6EMP | RF5EMP | RF4EMP | RF3EMP | RF2EMP | RF1EMP | RF0EMP |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.55 RSCANnFESTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 17 | Reserved | These bits are read as the value after reset. |
| 16 | CF8EMP | Transmit/Receive FIFO Buffer Empty Status Flag |
| 15 | CF7EMP | 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. |
| 14 | CF6EMP | (k = 0 to 8) |
| 13 | CF5EMP | |
| 12 | CF4EMP | |
| 11 | CF3EMP | |
| 10 | CF2EMP | |
| 9 | CF1EMP | |
| 8 | CF0EMP | |
| 7 | RF7EMP | Receive FIFO Buffer Empty Status Flag |
| 6 | RF6EMP | 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message (buffer empty). |
| 5 | RF5EMP | (x = 0 to 7) |
| 4 | RF4EMP | |
| 3 | RF3EMP | |
| 2 | RF2EMP | |
| 1 | RF1EMP | |
| 0 | RF0EMP | |

The RSCANnFESTS register is set to 03FF FFFF_H in global reset mode.

CFkEMP Flag (k = 0 to 8)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCANnCFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCANnRFSTStx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

17.3.9.2 RSCANnFFSTS — FIFO Full Status Register

Access: RSCANnFFSTS register can be read only in 32-bit units
 RSCANnFFSTSL, RSCANnFFSTSH registers can be read only in 16-bit units
 RSCANnFFSTSLL, RSCANnFFSTSLH, RSCANnFFSTSHL, RSCANnFFSTSHH registers can be read only in 8-bit units

Address: RSCANnFFSTS: <RSCFDn_base> + 023C_H
 RSCANnFFSTSL: <RSCFDn_base> + 023C_H, RSCANnFFSTSH: <RSCFDn_base> + 023E_H
 RSCANnFFSTSLL: <RSCFDn_base> + 023C_H, RSCANnFFSTSLH: <RSCFDn_base> + 023D_H,
 RSCANnFFSTSHL: <RSCFDn_base> + 023E_H, RSCANnFFSTSHH: <RSCFDn_base> + 023F_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CF8FLL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | CF7FLL | CF6FLL | CF5FLL | CF4FLL | CF3FLL | CF2FLL | CF1FLL | CF0FLL | RF7FLL | RF6FLL | RF5FLL | RF4FLL | RF3FLL | RF2FLL | RF1FLL | RF0FLL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.56 RSCANnFFSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 17 | Reserved | These bits are read as the value after reset. |
| 16 | CF8FLL | Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. |
| 15 | CF7FLL | |
| 14 | CF6FLL | (k = 0 to 8) |
| 13 | CF5FLL | |
| 12 | CF4FLL | |
| 11 | CF3FLL | |
| 10 | CF2FLL | |
| 9 | CF1FLL | |
| 8 | CF0FLL | |
| 7 | RF7FLL | Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. |
| 6 | RF6FLL | (x = 0 to 7) |
| 5 | RF5FLL | |
| 4 | RF4FLL | |
| 3 | RF3FLL | |
| 2 | RF2FLL | |
| 1 | RF1FLL | |
| 0 | RF0FLL | |

The RSCANnFFSTS register is cleared to 0000 0000_H in global reset mode.

CFkFLL Flag (k = 0 to 8)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCANnCFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).

When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLl Flag (x = 0 to 7)

The RFxFLl flag is set to 1 when the RFFLL flag in the RSCANnRFSTsx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLl flag is cleared to 0.

17.3.9.3 RSCANnFMSTS — FIFO Message Lost Status Register

Access: RSCANnFMSTS register can be read only in 32-bit units
 RSCANnFMSTSL, RSCANnFMSTSH registers can be read only in 16-bit units
 RSCANnFMSTSL, RSCANnFMSTSLH, RSCANnFMSTSHL, RSCANnFMSTSHH registers can be read only in 8-bit units

Address: RSCANnFMSTS: <RSCFDn_base> + 0240_H
 RSCANnFMSTSL: <RSCFDn_base> + 0240_H, RSCANnFMSTSH: <RSCFDn_base> + 0242_H
 RSCANnFMSTSL: <RSCFDn_base> + 0240_H, RSCANnFMSTSLH: <RSCFDn_base> + 0241_H,
 RSCANnFMSTSHL: <RSCFDn_base> + 0242_H, RSCANnFMSTSHH: <RSCFDn_base> + 0243_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CF8MLT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | CF7MLT | CF6MLT | CF5MLT | CF4MLT | CF3MLT | CF2MLT | CF1MLT | CF0MLT | RF7MLT | RF6MLT | RF5MLT | RF4MLT | RF3MLT | RF2MLT | RF1MLT | RF0MLT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.57 RSCANnFMSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 17 | Reserved | These bits are read as the value after reset. |
| 16 | CF8MLT | Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. |
| 15 | CF7MLT | |
| 14 | CF6MLT | (k = 0 to 8) |
| 13 | CF5MLT | |
| 12 | CF4MLT | |
| 11 | CF3MLT | |
| 10 | CF2MLT | |
| 9 | CF1MLT | |
| 8 | CF0MLT | |
| 7 | RF7MLT | Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. |
| 6 | RF6MLT | |
| 5 | RF5MLT | (x = 0 to 7) |
| 4 | RF4MLT | |
| 3 | RF3MLT | |
| 2 | RF2MLT | |
| 1 | RF1MLT | |
| 0 | RF0MLT | |

The RSCANnFMSTS register is cleared to 0000 0000_H in global reset mode.

CFkMLT Flag (k = 0 to 8)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCANnRFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

17.3.9.4 RSCANnRFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: RSCANnRFISTS register can be read only in 32-bit units
 RSCANnRFISTS_{SL}, RSCANnRFISTS_{SH} registers can be read only in 16-bit units
 RSCANnRFISTS_{SL}_L, RSCANnRFISTS_{SL}_H, RSCANnRFISTS_{SH}_L, RSCANnRFISTS_{SH}_H registers can be read only in 8-bit units

Address: RSCANnRFISTS: <RSCFDn_base> + 0244_H
 RSCANnRFISTS_{SL}: <RSCFDn_base> + 0244_H, RSCANnRFISTS_{SH}: <RSCFDn_base> + 0246_H
 RSCANnRFISTS_{SL}_L: <RSCFDn_base> + 0244_H, RSCANnRFISTS_{SL}_H: <RSCFDn_base> + 0245_H,
 RSCANnRFISTS_{SH}_L: <RSCFDn_base> + 0246_H, RSCANnRFISTS_{SH}_H: <RSCFDn_base> + 0247_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|-------|-------|-------|-------|-------|-------|-------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | RF7IF | RF6IF | RF5IF | RF4IF | RF3IF | RF2IF | RF1IF | RF0IF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.58 RSCANnRFISTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 8 | Reserved | These bits are read as the value after reset. |
| 7 | RF7IF | Receive FIFO Buffer Interrupt Request Status Flag |
| 6 | RF6IF | 0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. |
| 5 | RF5IF | (x = 0 to 7) |
| 4 | RF4IF | |
| 3 | RF3IF | |
| 2 | RF2IF | |
| 1 | RF1IF | |
| 0 | RF0IF | |

The RSCANnRFISTS register is cleared to 0000 0000_H in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RSCANnRFISTS_x register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

17.3.9.5 RSCANnCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RSCANnCFRISTS register can be read only in 32-bit units
 RSCANnCFRISTSL, RSCANnCFRISTSH registers can be read only in 16-bit units
 RSCANnCFRISTSLL, RSCANnCFRISTSLH, RSCANnCFRISTSHL, RSCANnCFRISTSHH registers can be read only in 8-bit units

Address: RSCANnCFRISTS: <RSCFDn_base> + 0248_H
 RSCANnCFRISTSL: <RSCFDn_base> + 0248_H, RSCANnCFRISTSH: <RSCFDn_base> + 024A_H
 RSCANnCFRISTSLL: <RSCFDn_base> + 0248_H, RSCANnCFRISTSLH: <RSCFDn_base> + 0249_H,
 RSCANnCFRISTSHL: <RSCFDn_base> + 024A_H, RSCANnCFRISTSHH: <RSCFDn_base> + 024B_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | CF8RXI | CF7RXI | CF6RXI | CF5RXI | CF4RXI | CF3RXI | CF2RXI | CF1RXI | CF0RXI |
| | | | | | | | | F | F | F | F | F | F | F | F | F |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.59 RSCANnCFRISTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 9 | Reserved | These bits are read as the value after reset. |
| 8 | CF8RXIF | Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 8) |
| 7 | CF7RXIF | |
| 6 | CF6RXIF | |
| 5 | CF5RXIF | |
| 4 | CF4RXIF | |
| 3 | CF3RXIF | |
| 2 | CF2RXIF | |
| 1 | CF1RXIF | |
| 0 | CF0RXIF | |

The RSCANnCFRISTS register is cleared to 0000 0000_H in global reset mode.

CFkRXIF Flag (k = 0 to 8)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

17.3.9.6 RSCANnCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RSCANnCFTISTS register can be read only in 32-bit units
 RSCANnCFTISTSL, RSCANnCFTISTSH registers can be read only in 16-bit units
 RSCANnCFTISTSLL, RSCANnCFTISTSLH, RSCANnCFTISTSHL, RSCANnCFTISTSHH registers can be read only in 8-bit units

Address: RSCANnCFTISTS: <RSCFDn_base> + 024C_H
 RSCANnCFTISTSL: <RSCFDn_base> + 024C_H, RSCANnCFTISTSH: <RSCFDn_base> + 024E_H
 RSCANnCFTISTSLL: <RSCFDn_base> + 024C_H, RSCANnCFTISTSLH: <RSCFDn_base> + 024D_H,
 RSCANnCFTISTSHL: <RSCFDn_base> + 024E_H, RSCANnCFTISTSHH: <RSCFDn_base> + 024F_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | — | — | — | — | — | — | — | CF8TXI | CF7TXI | CF6TXI | CF5TXI | CF4TXI | CF3TXI | CF2TXI | CF1TXI | CF0TXI |
| | | | | | | | | F | F | F | F | F | F | F | F | F |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.60 RSCANnCFTISTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 9 | Reserved | These bits are read as the value after reset. |
| 8 | CF8TXIF | Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 8) |
| 7 | CF7TXIF | |
| 6 | CF6TXIF | |
| 5 | CF5TXIF | |
| 4 | CF4TXIF | |
| 3 | CF3TXIF | |
| 2 | CF2TXIF | |
| 1 | CF1TXIF | |
| 0 | CF0TXIF | |

The RSCANnCFTISTS register is cleared to 0000 0000_H in global reset mode.

CFkTXIF Flag (k = 0 to 8)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

17.3.10 Details of Transmit Buffer-Related Registers

17.3.10.1 RSCANnTMCp — Transmit Buffer Control Register p (p = 0 to 47)

Access: RSCANnTMCp register can be read/written in 8-bit units

Address: RSCANnTMCp: <RSCFDn_base> + 0250_H + (01_H × p)

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | TMOM | TMTAR | TMTR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W*1 | R/W*1 |

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 17.61 RSCANnTMCp Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 3 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 2 | TMOM | One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled. |
| 1 | TMTAR | Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested. |
| 0 | TMTR | Transmit Request 0: Transmission is not requested. 1: Transmission is requested. |

When the RSCANnTMCp register meets any of the following conditions, set it to 00_H.

- The RSCANnTMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCANnCFCCk register (p = m × 16 + the value of CFTML[3:0] bits).
- The RSCANnTMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCANnTXQCCm (m = 0 to 2) register (p = (m × 16 + 15) to (m × 16 + 15 – the value of TXQDC[3:0] bits)).

All of the bits in the RSCANnTMCp register are cleared to 0 in channel reset mode. Modify the RSCANnTMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. If transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCANnTMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RSCANnTMSTSp register is 00_B.

17.3.10.2 RSCANnTMSTSp — Transmit Buffer Status Register p (p = 0 to 47)

Access: RSCANnTMSTSp register can be read/written in 8-bit units

Address: RSCANnTMSTSp: <RSCFDn_base> + 02D0_H + (01_H × p)

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|--------|-------|------------|-----|--------|
| | — | — | — | TMTARM | TMTRM | TMTRF[1:0] | | TMTSTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R |

Table 17.62 RSCANnTMSTSp Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 7 to 5 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 4 | TMTARM | Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present. |
| 3 | TMTRM | Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present. |
| 2, 1 | TMTRF[1:0] | Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request). |
| 0 | TMTSTS | Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress. |

All of the bits in the RSCANnTMSTSp register are cleared to 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCANnTMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCANnTMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCANnTMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCANnTMCp register is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCANnTMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCANnTMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

17.3.10.3 RSCANnTMIDp — Transmit Buffer ID Register p (p = 0 to 47)

Access: RSCANnTMIDp register can be read/written in 32-bit units
 RSCANnTMIDpL, RSCANnTMIDpH registers can be read/written in 16-bit units
 RSCANnTMIDpLL, RSCANnTMIDpLH, RSCANnTMIDpHL, RSCANnTMIDpHH registers can be read/written in 8-bit units

Address: RSCANnTMIDp: $\langle \text{RSCFDn_base} \rangle + 1000_{\text{H}} + (10_{\text{H}} \times p)$
 RSCANnTMIDpL: $\langle \text{RSCFDn_base} \rangle + 1000_{\text{H}} + (10_{\text{H}} \times p)$,
 RSCANnTMIDpH: $\langle \text{RSCFDn_base} \rangle + 1002_{\text{H}} + (10_{\text{H}} \times p)$
 RSCANnTMIDpLL: $\langle \text{RSCFDn_base} \rangle + 1000_{\text{H}} + (10_{\text{H}} \times p)$,
 RSCANnTMIDpLH: $\langle \text{RSCFDn_base} \rangle + 1001_{\text{H}} + (10_{\text{H}} \times p)$,
 RSCANnTMIDpHL: $\langle \text{RSCFDn_base} \rangle + 1002_{\text{H}} + (10_{\text{H}} \times p)$,
 RSCANnTMIDpHH: $\langle \text{RSCFDn_base} \rangle + 1003_{\text{H}} + (10_{\text{H}} \times p)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-------|-------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | TMIDE | TMRTR | THLEN | TMID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.63 RSCANnTMIDp Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 | TMIDE | Transmit Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | TMRTR | Transmit Buffer RTR 0: Data frame 1: Remote frame |
| 29 | THLEN | Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer. |
| 28 to 0 | TMID[28:0] | Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. |

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp (timestamp is included if the TMTSCE bit in the RSCANnGCFG register is 1)) of transmit messages is stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

17.3.10.4 RSCANnTMPTRp — Transmit Buffer Pointer Register p (p = 0 to 47)

Access: RSCANnTMPTRp register can be read/written in 32-bit units
 RSCANnTMPTRpL, RSCANnTMPTRpH registers can be read/written in 16-bit units
 RSCANnTMPTRpLL, RSCANnTMPTRpLH, RSCANnTMPTRpHL, RSCANnTMPTRpHH registers can be read/written in 8-bit units

Address: RSCANnTMPTRp: <RSCFDn_base> + 1004_H + (10_H × p)
 RSCANnTMPTRpL: <RSCFDn_base> + 1004_H + (10_H × p),
 RSCANnTMPTRpH: <RSCFDn_base> + 1006_H + (10_H × p)
 RSCANnTMPTRpLL: <RSCFDn_base> + 1004_H + (10_H × p),
 RSCANnTMPTRpLH: <RSCFDn_base> + 1005_H + (10_H × p),
 RSCANnTMPTRpHL: <RSCFDn_base> + 1006_H + (10_H × p),
 RSCANnTMPTRpHH: <RSCFDn_base> + 1007_H + (10_H × p)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-----|----|----|----|----|------------|-----|-----|-----|-----|-----|-----|-----|
| | TMDLC[3:0] | | | | — | — | — | — | TMPTR[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.64 RSCANnTMPTRp Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 28 | TMDLC[3:0] | Transmit Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes |
| 27 to 24 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 23 to 16 | TMPTR[7:0] | Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer. |
| 15 to 0 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCANnTMIDp register is set to 0 (data frame). If the data length is set to 1001_{B} or more, the transmit data is 8 bytes long.

When the TMRTR bit is set to 1 (remote frame), these bits set the data length of the message to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

17.3.10.5 RSCANnTMDF0_p — Transmit Buffer Data Field 0 Register p (p = 0 to 47)

Access: RSCANnTMDF0_p register can be read/written in 32-bit units
 RSCANnTMDF0_pL, RSCANnTMDF0_pH registers can be read/written in 16-bit units
 RSCANnTMDF0_pLL, RSCANnTMDF0_pLH, RSCANnTMDF0_pHL, RSCANnTMDF0_pHH registers can be read/written in 8-bit units

Address: RSCANnTMDF0_p: <RSCFDn_base> + 1008_H + (10_H × p)
 RSCANnTMDF0_pL: <RSCFDn_base> + 1008_H + (10_H × p),
 RSCANnTMDF0_pH: <RSCFDn_base> + 100A_H + (10_H × p)
 RSCANnTMDF0_pLL: <RSCFDn_base> + 1008_H + (10_H × p),
 RSCANnTMDF0_pLH: <RSCFDn_base> + 1009_H + (10_H × p),
 RSCANnTMDF0_pHL: <RSCFDn_base> + 100A_H + (10_H × p),
 RSCANnTMDF0_pHH: <RSCFDn_base> + 100B_H + (10_H × p)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| | TMDB3[7:0] | | | | | | | | TMDB2[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMDB1[7:0] | | | | | | | | TMDB0[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.65 RSCANnTMDF0_p Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------------|
| 31 to 24 | TMDB3[7:0] | Transmit Buffer Data Byte 3 |
| 23 to 16 | TMDB2[7:0] | Transmit Buffer Data Byte 2 |
| 15 to 8 | TMDB1[7:0] | Transmit Buffer Data Byte 1 |
| 7 to 0 | TMDB0[7:0] | Transmit Buffer Data Byte 0 |
| | | Set the transmit buffer data. |

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

17.3.10.6 RSCANnTMDF1_p — Transmit Buffer Data Field 1 Register p (p = 0 to 47)

Access: RSCANnTMDF1_p register can be read/written in 32-bit units
 RSCANnTMDF1_pL, RSCANnTMDF1_pH registers can be read/written in 16-bit units
 RSCANnTMDF1_pLL, RSCANnTMDF1_pLH, RSCANnTMDF1_pHL, RSCANnTMDF1_pHH registers can be read/written in 8-bit units

Address: RSCANnTMDF1_p: <RSCFDn_base> + 100C_H + (10_H × p)
 RSCANnTMDF1_pL: <RSCFDn_base> + 100C_H + (10_H × p),
 RSCANnTMDF1_pH: <RSCFDn_base> + 100E_H + (10_H × p)
 RSCANnTMDF1_pLL: <RSCFDn_base> + 100C_H + (10_H × p),
 RSCANnTMDF1_pLH: <RSCFDn_base> + 100D_H + (10_H × p),
 RSCANnTMDF1_pHL: <RSCFDn_base> + 100E_H + (10_H × p),
 RSCANnTMDF1_pHH: <RSCFDn_base> + 100F_H + (10_H × p)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| | TMDB7[7:0] | | | | | | | | TMDB6[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMDB5[7:0] | | | | | | | | TMDB4[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.66 RSCANnTMDF1_p Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------------|
| 31 to 24 | TMDB7[7:0] | Transmit Buffer Data Byte 7 |
| 23 to 16 | TMDB6[7:0] | Transmit Buffer Data Byte 6 |
| 15 to 8 | TMDB5[7:0] | Transmit Buffer Data Byte 5 |
| 7 to 0 | TMDB4[7:0] | Set the transmit buffer data. |

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

17.3.10.7 RSCANnTMIECy — Transmit Buffer Interrupt Enable Configuration Register y (y = 0, 1)

Access: RSCANnTMIECy register can be read/written in 32-bit units
 RSCANnTMIECyL, RSCANnTMIECyH registers can be read/written in 16-bit units
 RSCANnTMIECyLL, RSCANnTMIECyLH, RSCANnTMIECyHL, RSCANnTMIECyHH registers can be read/written in 8-bit units

Address: RSCANnTMIECy: <RSCFDn_base> + 0390_H + (04_H × y)
 RSCANnTMIECyL: <RSCFDn_base> + 0390_H + (04_H × y),
 RSCANnTMIECyH: <RSCFDn_base> + 0392_H + (04_H × y)
 RSCANnTMIECyLL: <RSCFDn_base> + 0390_H + (04_H × y),
 RSCANnTMIECyLH: <RSCFDn_base> + 0391_H + (04_H × y),
 RSCANnTMIECyHL: <RSCFDn_base> + 0392_H + (04_H × y),
 RSCANnTMIECyHH: <RSCFDn_base> + 0393_H + (04_H × y)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TMIEp (p = y × 32 + 31 to y × 32 + 16 (y = 0)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMIEp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.67 RSCANnTMIECy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | TMIEp | Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled When y = 1, these bits are reserved. When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | TMIEp | Transmit Buffer Interrupt Enable p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled. |

TMIEp Bits (p = 0 to 47)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCANnTMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

Table 17.68 shows the bit assignment.

Table 17.68 TMIEp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.3.11 Details of Transmit Buffer Status-Related Registers

17.3.11.1 RSCANnTMTRSTSy — Transmit Buffer Transmit Request Status Register y (y = 0, 1)

Access: RSCANnTMTRSTSy register can be read only in 32-bit units
RSCANnTMTRSTSyL, RSCANnTMTRSTSyH registers can be read only in 16-bit units
RSCANnTMTRSTSyLL, RSCANnTMTRSTSyLH, RSCANnTMTRSTSyHL, RSCANnTMTRSTSyHH registers can be read only in 8-bit units

Address: RSCANnTMTRSTSy: $\langle \text{RSCFDn_base} \rangle + 0350_{\text{H}} + (04_{\text{H}} \times y)$
RSCANnTMTRSTSyL: $\langle \text{RSCFDn_base} \rangle + 0350_{\text{H}} + (04_{\text{H}} \times y)$,
RSCANnTMTRSTSyH: $\langle \text{RSCFDn_base} \rangle + 0352_{\text{H}} + (04_{\text{H}} \times y)$
RSCANnTMTRSTSyLL: $\langle \text{RSCFDn_base} \rangle + 0350_{\text{H}} + (04_{\text{H}} \times y)$,
RSCANnTMTRSTSyLH: $\langle \text{RSCFDn_base} \rangle + 0351_{\text{H}} + (04_{\text{H}} \times y)$,
RSCANnTMTRSTSyHL: $\langle \text{RSCFDn_base} \rangle + 0352_{\text{H}} + (04_{\text{H}} \times y)$,
RSCANnTMTRSTSyHH: $\langle \text{RSCFDn_base} \rangle + 0353_{\text{H}} + (04_{\text{H}} \times y)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| TMTRSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMTRSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.69 RSCANnTMTRSTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | TMTRSTSp | Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit request is present. 1: A transmit request is present. When y = 1, these bits are reserved. When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | TMTRSTSp | Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present. |

TMTRSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTR bit in the RSCANnTMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 17.70 shows the bit assignment.

Table 17.70 TMTRSTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.3.11.2 RSCANnTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y (y = 0, 1)

Access: RSCANnTMTARSTSy register can be read only in 32-bit units
 RSCANnTMTARSTSyL, RSCANnTMTARSTSyH registers can be read only in 16-bit units
 RSCANnTMTARSTSyLL, RSCANnTMTARSTSyLH, RSCANnTMTARSTSyHL, RSCANnTMTARSTSyHH registers can be read only in 8-bit units

Address: RSCANnTMTARSTSy: <RSCFDn_base> + 0360_H + (04_H × y)
 RSCANnTMTARSTSyL: <RSCFDn_base> + 0360_H + (04_H × y),
 RSCANnTMTARSTSyH: <RSCFDn_base> + 0362_H + (04_H × y)
 RSCANnTMTARSTSyLL: <RSCFDn_base> + 0360_H + (04_H × y),
 RSCANnTMTARSTSyLH: <RSCFDn_base> + 0361_H + (04_H × y),
 RSCANnTMTARSTSyHL: <RSCFDn_base> + 0362_H + (04_H × y),
 RSCANnTMTARSTSyHH: <RSCFDn_base> + 0363_H + (04_H × y)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| TMTARSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMTARSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.71 RSCANnTMTARSTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | TMTARSTSp | Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit abort request is present. 1: A transmit abort request is present. When y = 1, these bits are reserved. When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | TMTARSTSp | Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present. |

TMTARSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTAR bit in the RSCANnTMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 17.72 shows the bit assignment.

Table 17.72 TMTARSTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.3.11.3 RSCANnTMCSTSy — Transmit Buffer Transmit Complete Status Register y (y = 0, 1)

Access: RSCANnTMCSTSy register can be read only in 32-bit units
RSCANnTMCSTSyL, RSCANnTMCSTSyH registers can be read only in 16-bit units
RSCANnTMCSTSyLL, RSCANnTMCSTSyLH, RSCANnTMCSTSyHL, RSCANnTMCSTSyHH registers can be read only in 8-bit units

Address: RSCANnTMCSTSy: <RSCFDn_base> + 0370_H + (04_H × y)
RSCANnTMCSTSyL: <RSCFDn_base> + 0370_H + (04_H × y),
RSCANnTMCSTSyH: <RSCFDn_base> + 0372_H + (04_H × y)
RSCANnTMCSTSyLL: <RSCFDn_base> + 0370_H + (04_H × y),
RSCANnTMCSTSyLH: <RSCFDn_base> + 0371_H + (04_H × y),
RSCANnTMCSTSyHL: <RSCFDn_base> + 0372_H + (04_H × y),
RSCANnTMCSTSyHH: <RSCFDn_base> + 0373_H + (04_H × y)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| TMCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.73 RSCANnTMCSTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | TMCSTSp | Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed. When y = 1, these bits are reserved. When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | TMCSTSp | Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed. |

TMCSTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCANnTMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMCSTSp flag is set to 1.

To clear the TMCSTSp flag to 0, set the corresponding TMTRF[1:0] flag to 00_B. This flag is cleared to 0 in channel reset mode.

Table 17.74 shows the bit assignment.

Table 17.74 TMCSTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.3.11.4 RSCANnTMTASTSy — Transmit Buffer Transmit Abort Status Register y (y = 0, 1)

Access: RSCANnTMTASTSy register can be read only in 32-bit units
 RSCANnTMTASTSyL, RSCANnTMTASTSyH registers can be read only in 16-bit units
 RSCANnTMTASTSyLL, RSCANnTMTASTSyLH, RSCANnTMTASTSyHL, RSCANnTMTASTSyHH registers can be read only in 8-bit units

Address: RSCANnTMTASTSy: <RSCFDn_base> + 0380_H + (04_H × y)
 RSCANnTMTASTSyL: <RSCFDn_base> + 0380_H + (04_H × y),
 RSCANnTMTASTSyH: <RSCFDn_base> + 0382_H + (04_H × y)
 RSCANnTMTASTSyLL: <RSCFDn_base> + 0380_H + (04_H × y),
 RSCANnTMTASTSyLH: <RSCFDn_base> + 0381_H + (04_H × y),
 RSCANnTMTASTSyHL: <RSCFDn_base> + 0382_H + (04_H × y),
 RSCANnTMTASTSyHH: <RSCFDn_base> + 0383_H + (04_H × y)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TMTASTSp (p = y × 32 + 31 to y × 32 × 16 (y = 0)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.75 RSCANnTMTASTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 16 | TMTASTSp | Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted When y = 1, these bits are reserved. When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | TMTASTSp | Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted. |

TMTASTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCANnTMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

To clear the TMTASTSp flag to 0, set the corresponding TMTRF[1:0] flag is set to 00_B. This flag is cleared to 0 in channel reset mode.

Table 17.76 shows the bit assignment.

Table 17.76 TMTASTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.3.12 Details of Transmit Queue-Related Registers

17.3.12.1 RSCANnTXQCCm — Transmit Queue Configuration and Control Register m (m = 0 to 2)

Access: RSCANnTXQCCm register can be read/written in 32-bit units
 RSCANnTXQCCmL, RSCANnTXQCCmH registers can be read/written in 16-bit units
 RSCANnTXQCCmLL, RSCANnTXQCCmLH, RSCANnTXQCCmHL, RSCANnTXQCCmHH registers can be read/written in 8-bit units

Address: RSCANnTXQCCm: <RSCFDn_base> + 03A0_H + (04_H × m)
 RSCANnTXQCCmL: <RSCFDn_base> + 03A0_H + (04_H × m),
 RSCANnTXQCCmH: <RSCFDn_base> + 03A2_H + (04_H × m)
 RSCANnTXQCCmLL: <RSCFDn_base> + 03A0_H + (04_H × m),
 RSCANnTXQCCmLH: <RSCFDn_base> + 03A1_H + (04_H × m),
 RSCANnTXQCCmHL: <RSCFDn_base> + 03A2_H + (04_H × m),
 RSCANnTXQCCmHH: <RSCFDn_base> + 03A3_H + (04_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|-------|-------|------------|-----|-----|-----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | TXQIM | TXQIE | TXQDC[3:0] | | | — | — | — | — | — | — | — | — | TXQE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R/W |

Table 17.77 RSCANnTXQCCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 14 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 13 | TXQIM | Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted. |
| 12 | TXQIE | Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled. |
| 11 to 8 | TXQDC[3:0] | Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1) transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited. |
| 7 to 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | TXQE | Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used. |

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from $(m \times 16 + 15)$ to $(m \times 16 + 0)$ (see **Table 17.16**). For examples of how buffer allocation is done, see **Figure 17.9**. Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 enables the transmit queue. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010_B or more.

17.3.12.2 RSCANnTXQSTSm — Transmit Queue Status Register m (m = 0 to 2)

Access: RSCANnTXQSTSm register can be read/written in 32-bit units
 RSCANnTXQSTSmL, RSCANnTXQSTSmH registers can be read/written in 16-bit units
 RSCANnTXQSTSmLL, RSCANnTXQSTSmLH, RSCANnTXQSTSmHL, RSCANnTXQSTSmHH registers can be read/written in 8-bit units

Address: RSCANnTXQSTSm: <RSCFDn_base> + 03C0_H + (04_H × m)
 RSCANnTXQSTSmL: <RSCFDn_base> + 03C0_H + (04_H × m),
 RSCANnTXQSTSmH: <RSCFDn_base> + 03C2_H + (04_H × m)
 RSCANnTXQSTSmLL: <RSCFDn_base> + 03C0_H + (04_H × m),
 RSCANnTXQSTSmLH: <RSCFDn_base> + 03C1_H + (04_H × m),
 RSCANnTXQSTSmHL: <RSCFDn_base> + 03C2_H + (04_H × m),
 RSCANnTXQSTSmHH: <RSCFDn_base> + 03C3_H + (04_H × m)

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|-------|------------|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQIF | TXQFL L | TXQEM P |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.78 RSCANnTXQSTSm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 13 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 12 to 8 | Reserved | When read, the undefined value is returned. When writing, write the value after reset. |
| 7 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | TXQIF | Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present. |
| 1 | TXQFL | Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full. |
| 0 | TXQEMP | Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty). |

TXQIF Flag

The TXQIF flag is set to 1 when the interrupt source specified by the TXQIM bit in the RSCANnTXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCANnTXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages stored in the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCANnTXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is stored in the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

17.3.12.3 RSCANnTXQPCTRm — Transmit Queue Pointer Control Register m (m = 0 to 2)

Access: RSCANnTXQPCTRm register can only be written in 32-bit units
 RSCANnTXQPCTRmL, RSCANnTXQPCTRmH registers can only be written in 16-bit units
 RSCANnTXQPCTRmLL, RSCANnTXQPCTRmLH, RSCANnTXQPCTRmHL, RSCANnTXQPCTRmHH registers can only be written in 8-bit units

Address: RSCANnTXQPCTRm: <RSCFDn_base> + 03E0_H + (04_H × m)
 RSCANnTXQPCTRmL: <RSCFDn_base> + 03E0_H + (04_H × m),
 RSCANnTXQPCTRmH: <RSCFDn_base> + 03E2_H + (04_H × m)
 RSCANnTXQPCTRmLL: <RSCFDn_base> + 03E0_H + (04_H × m),
 RSCANnTXQPCTRmLH: <RSCFDn_base> + 03E1_H + (04_H × m),
 RSCANnTXQPCTRmHL: <RSCFDn_base> + 03E2_H + (04_H × m),
 RSCANnTXQPCTRmHH: <RSCFDn_base> + 03E3_H + (04_H × m)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | TXQPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.79 RSCANnTXQPCTRm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 8 | Reserved | The write value should be the value after reset. |
| 7 to 0 | TXQPC[7:0] | Transmit Queue Pointer Control Writing FF _H to these bits moves the write pointer of the transmit queue to the next queue buffer. |

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request for the message. Write transmit messages to the RSCANnTMIDp, RSCANnTMPTRp, RSCANnTMDFO_p, and RSCANnTMDF1_p registers (p = 15, 31, 47) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCANnTXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCANnTXQSTSm register is 0 (The transmit queue is not full).

17.3.13 Details of Transmit history-related Registers

17.3.13.1 RSCANnTHLCCm — Transmit History Configuration and Control Register m (m = 0 to 2)

Access: RSCANnTHLCCm register can be read/written in 32-bit units
 RSCANnTHLCCmL, RSCANnTHLCCmH registers can be read/written in 16-bit units
 RSCANnTHLCCmLL, RSCANnTHLCCmLH, RSCANnTHLCCmHL, RSCANnTHLCCmHH registers can be read/written in 8-bit units

Address: RSCANnTHLCCm: <RSCFDn_base> + 0400_H + (04_H × m)
 RSCANnTHLCCmL: <RSCFDn_base> + 0400_H + (04_H × m),
 RSCANnTHLCCmH: <RSCFDn_base> + 0402_H + (04_H × m)
 RSCANnTHLCCmLL: <RSCFDn_base> + 0400_H + (04_H × m),
 RSCANnTHLCCmLH: <RSCFDn_base> + 0401_H + (04_H × m),
 RSCANnTHLCCmHL: <RSCFDn_base> + 0402_H + (04_H × m),
 RSCANnTHLCCmHH: <RSCFDn_base> + 0403_H + (04_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|---------|-------|-------|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | THL DTE | THLIM | THLIE | — | — | — | — | — | — | — | THLE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R | R | R/W |

Table 17.80 RSCANnTHLCCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 11 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 10 | THLDTE | Transmit History Target Buffer Select 0: Entries from transmit/receive FIFO buffers and transmit queue 1: Entries from transmit buffers, transmit/receive FIFO buffers, and transmit queue |
| 9 | THLIM | Transmit History Interrupt Source Select 0: When 12 items of data have been stored in the transmit history buffer 1: Each time transmit history data is stored in the transmit history buffer |
| 8 | THLIE | Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled. |
| 7 to 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | THLE | Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used. |

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit is set to 0.

THLE Bit

Setting this bit to 1 enables the transmit history buffer. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in channel reset mode.

17.3.13.2 RSCANnTHLSTSm — Transmit History Status Register m (m = 0 to 2)

Access: RSCANnTHLSTSm register can be read/written in 32-bit units
 RSCANnTHLSTSmL, RSCANnTHLSTSmH register can be read/written in 16-bit units
 RSCANnTHLSTSmLL, RSCANnTHLSTSmLH, RSCANnTHLSTSmHL, RSCANnTHLSTSmHH registers can be read/written in 8-bit units

Address: RSCANnTHLSTSm: <RSCFDn_base> + 0420_H + (04_H × m)
 RSCANnTHLSTSmL: <RSCFDn_base> + 0420_H + (04_H × m),
 RSCANnTHLSTSmH: <RSCFDn_base> + 0422_H + (04_H × m)
 RSCANnTHLSTSmLL: <RSCFDn_base> + 0420_H + (04_H × m),
 RSCANnTHLSTSmLH: <RSCFDn_base> + 0421_H + (04_H × m),
 RSCANnTHLSTSmHL: <RSCFDn_base> + 0422_H + (04_H × m),
 RSCANnTHLSTSmHH: <RSCFDn_base> + 0423_H + (04_H × m)

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|------------|----|----|---|---|---|---|---|---|-------|--------|--------|--------|
| | — | — | — | THLMC[4:0] | | | | — | — | — | — | — | THLIF | THLELT | THLFLL | THLEMP |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.81 RSCANnTHLSTSm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 13 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 12 to 8 | THLMC[4:0] | Transmit History Buffer Unread Data Counter These bits indicate the number of unread data stored in the transmit history buffer. |
| 7 to 4 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 3 | THLIF | Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present. |
| 2 | THLELT | Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred. |
| 1 | THLFLL | Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full. |
| 0 | THLEMP | Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty). |

THLMC[4:0] Bits

These bits indicate the number of unread data stored in the transmit history buffer.

These bits are cleared to 0 in channel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCANnTHLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 16 items of data have been stored in the transmit history buffer, and is cleared to 0 when the number of data stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCANnTHLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single item of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCANnTHLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

17.3.13.3 RSCANnTHLPCTRm — Transmit History Pointer Control Register m (m = 0 to 2)

Access: RSCANnTHLPCTRm register can only be written in 32-bit units
 RSCANnTHLPCTRmL, RSCANnTHLPCTRmH registers can only be written in 16-bit units
 RSCANnTHLPCTRmLL, RSCANnTHLPCTRmLH, RSCANnTHLPCTRmHL, RSCANnTHLPCTRmHH registers can only be written in 8-bit units

Address: RSCANnTHLPCTRm: <RSCFDn_base> + 0440_H + (04_H × m)
 RSCANnTHLPCTRmL: <RSCFDn_base> + 0440_H + (04_H × m),
 RSCANnTHLPCTRmH: <RSCFDn_base> + 0442_H + (04_H × m)
 RSCANnTHLPCTRmLL: <RSCFDn_base> + 0440_H + (04_H × m),
 RSCANnTHLPCTRmLH: <RSCFDn_base> + 0441_H + (04_H × m),
 RSCANnTHLPCTRmHL: <RSCFDn_base> + 0442_H + (04_H × m),
 RSCANnTHLPCTRmHH: <RSCFDn_base> + 0443_H + (04_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | THLPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.82 RSCANnTHLPCTRm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 8 | Reserved | When writing, write the value after reset. |
| 7 to 0 | THLPC[7:0] | Transmit History List Pointer Control Writing FF _H to these bits moves the read pointer to the next unread data in the transmit history buffer. |

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCANnTHLSTSm register is decremented by 1. Write FF_H to the THLPC[7:0] bits after reading from the RSCANnTHLACCm register.

When writing FF_H to these bits, make sure that the THLE bit in the RSCANnTHLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCANnTHLSTSm register is 0.

17.3.13.4 RSCANnTHLACCm — Transmit History Access Register m (m = 0 to 2)

Access: RSCANnTHLACCm register can be read only in 32-bit units
 RSCANnTHLACCmL, RSCANnTHLACCmH registers can be read only in 16-bit units
 RSCANnTHLACCmLL, RSCANnTHLACCmLH, RSCANnTHLACCmHL, RSCANnTHLACCmHH registers can be read only in 8-bit units

Address: RSCANnTHLACCm: $\langle \text{RSCFDn_base} \rangle + 1800_{\text{H}} + (04_{\text{H}} \times m)$
 RSCANnTHLACCmL: $\langle \text{RSCFDn_base} \rangle + 1800_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCANnTHLACCmH: $\langle \text{RSCFDn_base} \rangle + 1802_{\text{H}} + (04_{\text{H}} \times m)$
 RSCANnTHLACCmLL: $\langle \text{RSCFDn_base} \rangle + 1800_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCANnTHLACCmLH: $\langle \text{RSCFDn_base} \rangle + 1801_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCANnTHLACCmHL: $\langle \text{RSCFDn_base} \rangle + 1802_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCANnTHLACCmHH: $\langle \text{RSCFDn_base} \rangle + 1803_{\text{H}} + (04_{\text{H}} \times m)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|---------|----|----|---------|----|----|----|----|
| | TMTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TID[7:0] | | | | | | | — | BN[3:0] | | | BT[2:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.83 RSCANnTHLACCm Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | |
|--------------|------------|---|------------------------------|----|----|--|---|---|---|-----------------|---|---|---|------------------------------|---|---|---|----------------|
| 31 to 16 | TMTS[15:0] | Timestamp Data The timestamp data of stored data can be read. | | | | | | | | | | | | | | | | |
| 15 to 8 | TID[7:0] | Label Data The label information of stored data can be read. | | | | | | | | | | | | | | | | |
| 7 | Reserved | When read, the value after reset is returned. | | | | | | | | | | | | | | | | |
| 6 to 3 | BN[3:0] | Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read. | | | | | | | | | | | | | | | | |
| 2 to 0 | BT[2:0] | Buffer Type Data <table border="0"> <tr> <td>b2</td> <td>b1</td> <td>b0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Transmit buffer</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Transmit/receive FIFO buffer</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Transmit queue</td> </tr> </table> | b2 | b1 | b0 | | 0 | 0 | 1 | Transmit buffer | 0 | 1 | 0 | Transmit/receive FIFO buffer | 1 | 0 | 0 | Transmit queue |
| b2 | b1 | b0 | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | Transmit buffer | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | Transmit/receive FIFO buffer | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | Transmit queue | | | | | | | | | | | | | | | |

TMTS[15:0] Bits

When the TMTSCE bit in the RSCANnGCFG register is 1, timestamp values in transmit history data stored in the transmit history buffer are displayed. When the TMTSCE bit is 0, these bits are always read as 0.

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

17.3.14 Details of Test-Related Registers

17.3.14.1 RSCANnGTSTCFG — Global Test Configuration Register

Access: RSCANnGTSTCFG register can be read/written in 32-bit units
 RSCANnGTSTCFGL, RSCANnGTSTCFGH registers can be read/written in 16-bit units
 RSCANnGTSTCFGLL, RSCANnGTSTCFGLH, RSCANnGTSTCFGHL, RSCANnGTSTCFGHH registers can be read/written in 8-bit units

Address: RSCANnGTSTCFG: <RSCFDn_base> + 0468_H
 RSCANnGTSTCFGL: <RSCFDn_base> + 0468_H, RSCANnGTSTCFGH: <RSCFDn_base> + 046A_H
 RSCANnGTSTCFGLL: <RSCFDn_base> + 0468_H, RSCANnGTSTCFGLH: <RSCFDn_base> + 0469_H,
 RSCANnGTSTCFGHL: <RSCFDn_base> + 046A_H, RSCANnGTSTCFGHH: <RSCFDn_base> + 046B_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|------------|-----|-----|-----|---------|---------|---------|
| | — | — | — | — | — | — | — | — | — | RTMPS[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | C2ICBCE | C1ICBCE | C0ICBCE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 17.84 RSCANnGTSTCFG Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 23 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 22 to 16 | RTMPS[6:0] | RAM Test Page Configuration Set a value within a range of page 0 (00 _H) to page 29 (1D _H). |
| 15 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | C2ICBCE | CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled 1: CAN2 inter-channel communication test is enabled. |
| 1 | C1ICBCE | CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled. |
| 0 | C0ICBCE | CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled. |

Modify the RSCANnGTSTCFG register only in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to 1D_H, inclusive.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

17.3.14.2 RSCANnGTSTCTR — Global Test Control Register

Access: RSCANnGTSTCTR register can be read/written in 32-bit units
 RSCANnGTSTCTRL, RSCANnGTSTCTRH registers can be read/written in 16-bit units
 RSCANnGTSTCTRLL, RSCANnGTSTCTRLH, RSCANnGTSTCTRHL, RSCANnGTSTCTRHH registers can be read/written in 8-bit units

Address: RSCANnGTSTCTR: <RSCFDn_base> + 046C_H
 RSCANnGTSTCTRL: <RSCFDn_base> + 046C_H, RSCANnGTSTCTRH: <RSCFDn_base> + 046E_H
 RSCANnGTSTCTRLL: <RSCFDn_base> + 046C_H, RSCANnGTSTCTRLH: <RSCFDn_base> + 046D_H,
 RSCANnGTSTCTRHL: <RSCFDn_base> + 046E_H, RSCANnGTSTCTRHH: <RSCFDn_base> + 046F_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|---------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RTME | — | ICBCTME |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R | R/W |

Table 17.85 RSCANnGTSTCTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | RTME | RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled. |
| 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | ICBCTME | Inter-Channel Communication Test Enable 0: Inter-channel communication test is disabled 1: Inter-channel communication test is enabled |

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode (see **Figure 17.37**).

1. Set the GMDC[1:0] bits in the RSCANnGCTR register to 10_B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 2) in the RSCANnGTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

This bit is cleared to 0 in global reset mode.

17.3.14.3 RSCANnGLOCKK — Global Lock Key Register

Access: RSCANnGLOCKK register can be write only in 32-bit units.
RSCANnGLOCKKL, RSCANnGLOCKKH registers can be write only in 16-bit units.

Address: RSCANnGLOCKK: <RSCFDn_base> + 047C_H
RSCANnGLOCKKL: <RSCFDn_base> + 047C_H, RSCANnGLOCKKH: <RSCFDn_base> + 047E_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LOCK[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 |

Note 1. Writing to these bits is effective only when the RS-CANFD module is in global test mode.

Table 17.86 RSCANnGLOCKK Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 16 | Reserved | When writing these bits, write the value after reset. |
| 15 to 0 | LOCK[15:0] | Lock Key These bits are key bits to release protection of test mode. |

The RSCANnGLOCKK register releases protection of special test bits and is write only.

For the protection release data, see **Section 17.11.4.2, Procedure for Releasing the Protection.**

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCANnGTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCFDn_base> + 0000_H to <RSCFDn_base> + 04FF_H) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

17.3.14.4 RSCANnRPGACC_r— RAM Test Page Access Register *r* (*r* = 0 to 63)

Access: RSCANnRPGACC_r register can be read/written in 32-bit units
 RSCANnRPGACC_{rL}, RSCANnRPGACC_{rH} registers can be read/written in 16-bit units
 RSCANnRPGACC_{rLL}, RSCANnRPGACC_{rLH}, RSCANnRPGACC_{rHL}, RSCANnRPGACC_{rHH} registers can be read/written in 8-bit units

Address: RSCANnRPGACC_r: <RSCFDn_base> + 1900_H + (04_H × *r*)
 RSCANnRPGACC_{rL}: <RSCFDn_base> + 1900_H + (04_H × *r*),
 RSCANnRPGACC_{rH}: <RSCFDn_base> + 1902_H + (04_H × *r*)
 RSCANnRPGACC_{rLL}: <RSCFDn_base> + 1900_H + (04_H × *r*),
 RSCANnRPGACC_{rLH}: <RSCFDn_base> + 1901_H + (04_H × *r*),
 RSCANnRPGACC_{rHL}: <RSCFDn_base> + 1902_H + (04_H × *r*),
 RSCANnRPGACC_{rHH}: <RSCFDn_base> + 1903_H + (04_H × *r*)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RDTA[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RDTA[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.87 RSCANnRPGACC_r Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 0 | RDTA [31:0] | RAM Data Test Access Data can be read and written in RSCAN RAM. |

Modify the RSCANnRPGACC_r register in global test mode with the RTME bit in the RSCANnGTSTCTR register set to 1 (RAM test is enabled).

The RSCANnRPGACC_r register can be read and written when the RTME bit is set to 1.

17.3.15 Details of Mode Read Register

17.3.15.1 RSCANnCANFDMDR — CAN FD Mode Read Register

Access: This register can be read in 32-bit units

Address: <RSCFDn_base> + 8000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FDMDR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.88 RSCANnCANFDMDR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 0 | FDMDR | CAN FD mode read 0: classical CAN mode is selected. 1: CAN FD mode is selected. |

17.4 Registers (CAN FD Mode)

This section describes all registers to be used when the RS-CANFD is used in CAN FD mode.

17.4.1 List of Registers

The following tables list RS-CANFD registers to be used in CAN FD mode.

For details about <RSCFDn_base>, see **Section 17.1.2, Register Base Address**.

For details about registers initialized in Global reset mode or Channel reset mode, see following.

- **Table 17.180, Registers Initialized in Global Reset Mode or Channel Reset Mode**
- **Table 17.181, Registers Initialized Only in Global Reset Mode**

Table 17.89 Registers (1/3)

| Module Name | Register Name | Symbol | Address |
|---|---|--------------------|---|
| Interface mode-related registers | | | |
| RSCFDn | Global interface mode select register | RSCFDnCFDGRMCFG | <RSCFDn_base> + 04FC _H |
| Channel-related registers | | | |
| RSCFDn | Channel m nominal bit rate configuration register | RSCFDnCFDCmNCFG | <RSCFDn_base> + 0000 _H + (10 _H × m) |
| RSCFDn | Channel m control register | RSCFDnCFDCmCTR | <RSCFDn_base> + 0004 _H + (10 _H × m) |
| RSCFDn | Channel m status register | RSCFDnCFDCmSTS | <RSCFDn_base> + 0008 _H + (10 _H × m) |
| RSCFDn | Channel m error flag register | RSCFDnCFDCmERFL | <RSCFDn_base> + 000C _H + (10 _H × m) |
| RSCFDn | Channel m data bit rate configuration register | RSCFDnCFDCmDCFG | <RSCFDn_base> + 0500 _H + (20 _H × m) |
| RSCFDn | Channel m CAN FD configuration register | RSCFDnCFDCmFDCFG | <RSCFDn_base> + 0504 _H + (20 _H × m) |
| RSCFDn | Channel m CAN FD control register | RSCFDnCFDCmFDCTR | <RSCFDn_base> + 0508 _H + (20 _H × m) |
| RSCFDn | Channel m CAN FD status register | RSCFDnCFDCmFDSTS | <RSCFDn_base> + 050C _H + (20 _H × m) |
| RSCFDn | Channel m CAN FD CRC register | RSCFDnCFDCmFDCRC | <RSCFDn_base> + 0510 _H + (20 _H × m) |
| Global-related registers | | | |
| RSCFDn | Global configuration register | RSCFDnCFDGCFCG | <RSCFDn_base> + 0084 _H |
| RSCFDn | Global control register | RSCFDnCFDGCCTR | <RSCFDn_base> + 0088 _H |
| RSCFDn | Global status register | RSCFDnCFDGCSTS | <RSCFDn_base> + 008C _H |
| RSCFDn | Global error flag register | RSCFDnCFDGERFL | <RSCFDn_base> + 0090 _H |
| RSCFDn | Global timestamp counter register | RSCFDnCFDGTSC | <RSCFDn_base> + 0094 _H |
| RSCFDn | Global TX Interrupt Status Register 0 | RSCFDnCFDGTINTSTS0 | <RSCFDn_base> + 0460 _H |
| RSCFDn | Global FD configuration register | RSCFDnCFDGFDCFCG | <RSCFDn_base> + 0474 _H |
| Receive rule-related registers | | | |
| RSCFDn | Receive Rule Entry Control Register | RSCFDnCFDGAFLLECTR | <RSCFDn_base> + 0098 _H |
| RSCFDn | Receive Rule Configuration Register 0 | RSCFDnCFDGAFLCFG0 | <RSCFDn_base> + 009C _H |
| RSCFDn | Receive Rule ID Register j | RSCFDnCFDGAFLIDj | <RSCFDn_base> + 1000 _H + (10 _H × j) |
| RSCFDn | Receive Rule Mask Register j | RSCFDnCFDGAFLMj | <RSCFDn_base> + 1004 _H + (10 _H × j) |
| RSCFDn | Receive Rule Pointer 0 Register j | RSCFDnCFDGAFLP0_j | <RSCFDn_base> + 1008 _H + (10 _H × j) |
| RSCFDn | Receive Rule Pointer 1 Register j | RSCFDnCFDGAFLP1_j | <RSCFDn_base> + 100C _H + (10 _H × j) |
| Receive buffer-related registers | | | |
| RSCFDn | Receive Buffer Number Register | RSCFDnCFDRMNB | <RSCFDn_base> + 00A4 _H |
| RSCFDn | Receive Buffer New Data Register y | RSCFDnCFDRMNDy | <RSCFDn_base> + 00A8 _H + (04 _H × y) |
| RSCFDn | Receive Buffer ID Register q | RSCFDnCFDRMIDq | <RSCFDn_base> + 2000 _H + (20 _H × q) |
| RSCFDn | Receive Buffer Pointer Register q | RSCFDnCFDRMPTRq | <RSCFDn_base> + 2004 _H + (20 _H × q) |
| RSCFDn | Receive buffer CAN FD status register q | RSCFDnCFDRMFDSTSq | <RSCFDn_base> + 2008 _H + (20 _H × q) |
| RSCFDn | Receive Buffer Data Field b Register q | RSCFDnCFDRMDFb_q | <RSCFDn_base> + 200C _H + (04 _H × b) + (20 _H × q) |

Table 17.89 Registers (2/3)

| Module Name | Register Name | Symbol | Address |
|---|--|--------------------|---|
| Receive FIFO buffer-related registers | | | |
| RSCFDn | Receive FIFO Buffer Configuration and Control Register x | RSCFDnCFDRFCCx | <RSCFDn_base> + 00B8 _H + (04 _H × x) |
| RSCFDn | Receive FIFO Buffer Status Register x | RSCFDnCFDRFSTSx | <RSCFDn_base> + 00D8 _H + (04 _H × x) |
| RSCFDn | Receive FIFO Buffer Pointer Control Register x | RSCFDnCFDRFPCTRx | <RSCFDn_base> + 00F8 _H + (04 _H × x) |
| RSCFDn | Receive FIFO Buffer Access ID Register x | RSCFDnCFDRFIDx | <RSCFDn_base> + 3000 _H + (80 _H × x) |
| RSCFDn | Receive FIFO Buffer Access Pointer Register x | RSCFDnCFDRFPTRx | <RSCFDn_base> + 3004 _H + (80 _H × x) |
| RSCFDn | Receive FIFO CAN FD status register x | RSCFDnCFDRFFDSTSx | <RSCFDn_base> + 3008 _H + (80 _H × x) |
| RSCFDn | Receive FIFO Buffer Access Data Field d Register x | RSCFDnCFDRFDf_d_x | <RSCFDn_base> + 300C _H + (04 _H × d) + (80 _H × x) |
| Transmit/Receive FIFO buffer related registers | | | |
| RSCFDn | Transmit/receive FIFO Buffer Configuration and Control Register k | RSCFDnCFDCFCCK | <RSCFDn_base> + 0118 _H + (04 _H × k) |
| RSCFDn | Transmit/receive FIFO Buffer Status Register k | RSCFDnCFDCFSTSk | <RSCFDn_base> + 0178 _H + (04 _H × k) |
| RSCFDn | Transmit/receive FIFO Buffer Pointer Control Register k | RSCFDnCFDCFPCTRk | <RSCFDn_base> + 01D8 _H + (04 _H × k) |
| RSCFDn | Transmit/receive FIFO Buffer Access ID Register k | RSCFDnCFDCFIDk | <RSCFDn_base> + 3400 _H + (80 _H × k) |
| RSCFDn | Transmit/receive FIFO Buffer Access Pointer Register k | RSCFDnCFDCFPTRk | <RSCFDn_base> + 3404 _H + (80 _H × k) |
| RSCFDn | Transmit/receive FIFO CAN FD configuration/status register k | RSCFDnCFDCFFDCSTSk | <RSCFDn_base> + 3408 _H + (80 _H × k) |
| RSCFDn | Transmit/receive FIFO Buffer Access Data Field d Register k | RSCFDnCFDCFDf_d_k | <RSCFDn_base> + 340C _H + (04 _H × d) + (80 _H × k) |
| FIFO status-related registers | | | |
| RSCFDn | FIFO Empty Status Register | RSCFDnCFDFESTS | <RSCFDn_base> + 0238 _H |
| RSCFDn | FIFO Full Status Register | RSCFDnCFDFFSTS | <RSCFDn_base> + 023C _H |
| RSCFDn | FIFO Message Lost Status Register | RSCFDnCFDFMSTS | <RSCFDn_base> + 0240 _H |
| RSCFDn | Receive FIFO Buffer Interrupt Flag Status Register | RSCFDnCFDRFISTS | <RSCFDn_base> + 0244 _H |
| RSCFDn | Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register | RSCFDnCFDCFRISTS | <RSCFDn_base> + 0248 _H |
| RSCFDn | Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register | RSCFDnCFDCFTISTS | <RSCFDn_base> + 024C _H |
| FIFO DMA-related registers | | | |
| RSCFDn | DMA enable register | RSCFDnCFDCDTCT | <RSCFDn_base> + 0490 _H |
| RSCFDn | DMA status register | RSCFDnCFDCDTSTS | <RSCFDn_base> + 0494 _H |
| Transmit buffer-related registers | | | |
| RSCFDn | Transmit Buffer Control Register p | RSCFDnCFDTMCp | <RSCFDn_base> + 0250 _H + (01 _H × p) |
| RSCFDn | Transmit Buffer Status Register p | RSCFDnCFDTMSTSp | <RSCFDn_base> + 02D0 _H + (01 _H × p) |
| RSCFDn | Transmit Buffer ID Register p | RSCFDnCFDTMIDp | <RSCFDn_base> + 4000 _H + (20 _H × p) |
| RSCFDn | Transmit Buffer Pointer Register p | RSCFDnCFDTMPTRp | <RSCFDn_base> + 4004 _H + (20 _H × p) |
| RSCFDn | Transmit buffer CAN FD configuration register p | RSCFDnCFDTMFDCRTP | <RSCFDn_base> + 4008 _H + (20 _H × p) |
| RSCFDn | Transmit Buffer Data Field b Register p | RSCFDnCFDTMDFb_p | <RSCFDn_base> + 400C _H + (04 _H × b) + (20 _H × p) |
| RSCFDn | Transmit Buffer Interrupt Enable Configuration Register y | RSCFDnCFDTMIECy | <RSCFDn_base> + 0390 _H + (04 _H × y) |
| Transmit buffer status-related registers | | | |
| RSCFDn | Transmit Buffer Transmit Request Status Register y | RSCFDnCFDTMTRSTSy | <RSCFDn_base> + 0350 _H + (04 _H × y) |
| RSCFDn | Transmit Buffer Transmit Abort Request Status Register y | RSCFDnCFDTMTARSTSy | <RSCFDn_base> + 0360 _H + (04 _H × y) |
| RSCFDn | Transmit Buffer Transmit Complete Status Register y | RSCFDnCFDTMTCSTSy | <RSCFDn_base> + 0370 _H + (04 _H × y) |
| RSCFDn | Transmit Buffer Transmit Abort Status Register y | RSCFDnCFDTMTASTSy | <RSCFDn_base> + 0380 _H + (04 _H × y) |
| Transmit queue-related registers | | | |
| RSCFDn | Transmit Queue Configuration and Control Register m | RSCFDnCFDTXQCCm | <RSCFDn_base> + 03A0 _H + (04 _H × m) |
| RSCFDn | Transmit Queue Status Register m | RSCFDnCFDTXQSTSm | <RSCFDn_base> + 03C0 _H + (04 _H × m) |
| RSCFDn | Transmit Queue Pointer Control Register m | RSCFDnCFDTXQPCTRM | <RSCFDn_base> + 03E0 _H + (04 _H × m) |

Table 17.89 Registers (3/3)

| Module Name | Register Name | Symbol | Address |
|---|---|-------------------|---|
| Transmit history-related registers | | | |
| RSCFDn | Transmit History Configuration and Control Register m | RSCFDnCFDTHLCCm | <RSCFDn_base> + 0400 _H + (04 _H × m) |
| RSCFDn | Transmit History Status Register m | RSCFDnCFDTHLSTSm | <RSCFDn_base> + 0420 _H + (04 _H × m) |
| RSCFDn | Transmit History Pointer Control Register m | RSCFDnCFDTHLPCTRm | <RSCFDn_base> + 0440 _H + (04 _H × m) |
| RSCFDn | Transmit History Access Register m | RSCFDnCFDTHLACCm | <RSCFDn_base> + 6000 _H + (04 _H × m) |
| Test-related registers | | | |
| RSCFDn | Global Test Configuration Register | RSCFDnCFDGTSTCFG | <RSCFDn_base> + 0468 _H |
| RSCFDn | Global Test Control Register | RSCFDnCFDGTSTCTR | <RSCFDn_base> + 046C _H |
| RSCFDn | Global Lock Key Register | RSCFDnCFDGLCKK | <RSCFDn_base> + 047C _H |
| RSCFDn | RAM Test Page Access Register r | RSCFDnCFDRPGACCr | <RSCFDn_base> + 6400 _H + (04 _H × r) |
| Mode read registers | | | |
| RSCFDn | CAN FD Mode Read Register | RSCFDnCANFMDR | <RSCFDn_base> + 8000 _H |

Table 17.90 Transmit Buffer p Allocated to Each Channel

| CANm | |
|-----------------------------|-----------------------------|
| Transmit buffer p | Transmit buffer 16 × m + 0 |
| | Transmit buffer 16 × m + 1 |
| | Transmit buffer 16 × m + 2 |
| | Transmit buffer 16 × m + 3 |
| | Transmit buffer 16 × m + 4 |
| | Transmit buffer 16 × m + 5 |
| | Transmit buffer 16 × m + 6 |
| | Transmit buffer 16 × m + 7 |
| | Transmit buffer 16 × m + 8 |
| | Transmit buffer 16 × m + 9 |
| | Transmit buffer 16 × m + 10 |
| | Transmit buffer 16 × m + 11 |
| | Transmit buffer 16 × m + 12 |
| | Transmit buffer 16 × m + 13 |
| | Transmit buffer 16 × m + 14 |
| Transmit buffer 16 × m + 15 | |

Table 17.91 Transmit/Receive FIFO Buffer k Allocated to Each Channel

| CANm | |
|-----------------------------------|--|
| Transmit/receive FIFO buffer k | Transmit/receive FIFO buffer 3 × m + 0 |
| | Transmit/receive FIFO buffer 3 × m + 1 |
| | Transmit/receive FIFO buffer 3 × m + 2 |

Table 17.92 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

| Setting of Bits CFTML[3:0] | Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer |
|----------------------------|--|
| 0000 _B | Transmit buffer 16 × m + 0 |
| 0001 _B | Transmit buffer 16 × m + 1 |
| 0010 _B | Transmit buffer 16 × m + 2 |
| 0011 _B | Transmit buffer 16 × m + 3 |
| 0100 _B | Transmit buffer 16 × m + 4 |
| 0101 _B | Transmit buffer 16 × m + 5 |
| 0110 _B | Transmit buffer 16 × m + 6 |
| 0111 _B | Transmit buffer 16 × m + 7 |
| 1000 _B | Transmit buffer 16 × m + 8 |
| 1001 _B | Transmit buffer 16 × m + 9 |
| 1010 _B | Transmit buffer 16 × m + 10 |
| 1011 _B | Transmit buffer 16 × m + 11 |
| 1100 _B | Transmit buffer 16 × m + 12 |
| 1101 _B | Transmit buffer 16 × m + 13 |
| 1110 _B | Transmit buffer 16 × m + 14 |
| 1111 _B | Transmit buffer 16 × m + 15 |

Table 17.93 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

| Setting of Bits TXQDC[3:0] | Transmit Buffer p Allocated to the Transmit Queue |
|----------------------------|---|
| 0000 _B | Setting prohibited |
| 0001 _B | Setting prohibited |
| 0010 _B | Transmit buffer 16 × m + 15 to 16 × m + 13 |
| 0011 _B | Transmit buffer 16 × m + 15 to 16 × m + 12 |
| 0100 _B | Transmit buffer 16 × m + 15 to 16 × m + 11 |
| 0101 _B | Transmit buffer 16 × m + 15 to 16 × m + 10 |
| 0110 _B | Transmit buffer 16 × m + 15 to 16 × m + 9 |
| 0111 _B | Transmit buffer 16 × m + 15 to 16 × m + 8 |
| 1000 _B | Transmit buffer 16 × m + 15 to 16 × m + 7 |
| 1001 _B | Transmit buffer 16 × m + 15 to 16 × m + 6 |
| 1010 _B | Transmit buffer 16 × m + 15 to 16 × m + 5 |
| 1011 _B | Transmit buffer 16 × m + 15 to 16 × m + 4 |
| 1100 _B | Transmit buffer 16 × m + 15 to 16 × m + 3 |
| 1101 _B | Transmit buffer 16 × m + 15 to 16 × m + 2 |
| 1110 _B | Transmit buffer 16 × m + 15 to 16 × m + 1 |
| 1111 _B | Transmit buffer 16 × m + 15 to 16 × m + 0 |

17.4.2 Details of Interface Mode-Related Registers

17.4.2.1 RSCFDnCFDGRMCFG — Global Interface Mode Select Register

Access: RSCFDnCFDGRMCFG register can be read/written in 32-bit units
 RSCFDnCFDGRMCFGL, RSCFDnCFDGRMCFGH registers can be read/written in 16-bit units
 RSCFDnCFDGRMCFGLL, RSCFDnCFDGRMCFGLH, RSCFDnCFDGRMCFGHL,
 RSCFDnCFDGRMCFGHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGRMCFG: <RSCFDn_base> + 04FC_H
 RSCFDnCFDGRMCFGL: <RSCFDn_base> + 04FC_H,
 RSCFDnCFDGRMCFGH: <RSCFDn_base> + 04FE_H
 RSCFDnCFDGRMCFGLL: <RSCFDn_base> + 04FC_H,
 RSCFDnCFDGRMCFGLH: <RSCFDn_base> + 04FD_H,
 RSCFDnCFDGRMCFGHL: <RSCFDn_base> + 04FE_H,
 RSCFDnCFDGRMCFGHH: <RSCFDn_base> + 04FF_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RCMC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 17.94 RSCFDnCFDGRMCFG Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 0 | RCMC | Interface Mode Select 0: Classical CAN mode 1: CAN FD mode |

Note 1. The RSCANnGRMCFG register and the RSCFDnCFDGRMCFG register are the same register. Therefore, set either one of the registers.

Modify the RSCFDnCFDGRMCFG register only in global reset mode. Before setting other RS-CANFD registers, set this register.

RCMC Bit

Setting this bit to 0 makes classical CAN mode available. Setting this bit to 1 causes the RS-CANFD module to transition to CAN FD mode. To switch the RS-CAN FD module from classical CAN mode to CAN FD mode, set the values after reset to all respective registers and bits allocated only to the register map in classical CAN mode and then modify the value of RSCFDnCFDGRMCFG register.

17.4.3 Details of Channel-Related Registers

17.4.3.1 RSCFDnCFDCmNCFG — Channel m Nominal Bit Rate Configuration Register (m = 0 to 2)

Access: RSCFDnCFDCmNCFG register can be read/written in 32-bit units
 RSCFDnCFDCmNCFG, RSCFDnCFDCmNCFGH registers can be read/written in 16-bit units
 RSCFDnCFDCmNCFGLL, RSCFDnCFDCmNCFGHL, RSCFDnCFDCmNCFGHLL, RSCFDnCFDCmNCFGHLH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmNCFG: <RSCFDn_base> + 0000_H + (10_H × m)
 RSCFDnCFDCmNCFG: <RSCFDn_base> + 0000_H + (10_H × m),
 RSCFDnCFDCmNCFGH: <RSCFDn_base> + 0002_H + (10_H × m)
 RSCFDnCFDCmNCFGLL: <RSCFDn_base> + 0000_H + (10_H × m),
 RSCFDnCFDCmNCFGHL: <RSCFDn_base> + 0001_H + (10_H × m),
 RSCFDnCFDCmNCFGHLL: <RSCFDn_base> + 0002_H + (10_H × m),
 RSCFDnCFDCmNCFGHLH: <RSCFDn_base> + 0003_H + (10_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|-----|-------------|-----|-----------|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | NTSEG2[4:0] | | | | — | NTSEG1[6:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NSJW[4:0] | | | | — | NBRP[9:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.95 RSCFDnCFDCmNCFG Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 29 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 28 to 24 | NTSEG2[4:0] | Nominal Bit Rate Time Segment 2 Control b28 b27 b26 b25 b24 0 0 0 0 0: Setting prohibited 0 0 0 0 1: 2 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq |
| 23 | Reserved | This bit is read as the value after reset. The write value should be the value after reset. |
| 22 to 16 | NTSEG1[6:0] | Nominal Bit Rate Time Segment 1 Control b22 b21 b20 b19 b18 b17 b16 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 1: Setting prohibited 0 0 0 0 0 1 0: Setting prohibited 0 0 0 0 0 1 1: 4 Tq : : 1 1 1 1 1 1 0: 127 Tq 1 1 1 1 1 1 1: 128 Tq |

Table 17.95 RSCFDnCFDCmNCFG Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 15 to 11 | NSJW[4:0] | Nominal Bit Rate Resynchronization Jump Width Control b15 b14 b13 b12 b11 0 0 0 0 0: 1 Tq 0 0 0 0 1: 2 Tq 0 0 0 1 0: 3 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq |
| 10 | Reserved | This bit is read as the value after reset. The write value should be the value after reset. |
| 9 to 0 | NBRP[9:0] | Nominal Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 1023), the nominal bit rate prescaler divides fCAN by (P + 1). |

Modify the RSCFDnCFDCmNCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. For the description and settings for bit timing parameters, see **Section 17.11.1, Initial Settings**.

NTSEG2[4:0] Bits

These bits specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of nominal bit rate.

Possible values are 2 to 32 Tq.

Set a value smaller than the value of the NTSEG1[6:0] bits.

NTSEG1[6:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of nominal bit rate as a Tq value.

Possible values are 4 to 128 Tq.

NSJW[4:0] Bits

These bits specify the resynchronization jump width of nominal bit rate as a Tq value. Possible values are 1 to 32 Tq. Specify a value equal to or smaller than the NTSEG2[4:0] value.

NBRP[9:0] Bits

The clock obtained by dividing the CAN clock (fCAN) by the nominal bit rate prescaler ((NBRP[9:0]) + 1) becomes CANmTq(N) clock (fCANTQ(N)m). One clock of the CANmTq(N) clock becomes one Time Quantum (Tq).

Be sure to specify the same value for both NBRP[9:0] and DBRP[7:0].

To specify a different value for the nominal bit rate and the data bit rate, change the values of the RSCFDnCFDCmNCFG.NTSEG1 and NTSEG2 bits and RSCFDnCFDCmDCFG.DTSEG1 and DTSEG2 bits, respectively.

17.4.3.2 RSCFDnCFDCmCTR — Channel m Control Register (m = 0 to 2)

Access: RSCFDnCFDCmCTR register can be read/written in 32-bit units
 RSCFDnCFDCmCTRL, RSCFDnCFDCmCTRHL registers can be read/written in 16-bit units
 RSCFDnCFDCmCTRLL, RSCFDnCFDCmCTRHL, RSCFDnCFDCmCTRHL, RSCFDnCFDCmCTRHL registers can be read/written in 8-bit units

Address: RSCFDnCFDCmCTR: <RSCFDn_base> + 0004_H + (10_H × m)
 RSCFDnCFDCmCTRL: <RSCFDn_base> + 0004_H + (10_H × m),
 RSCFDnCFDCmCTRHL: <RSCFDn_base> + 0006_H + (10_H × m)
 RSCFDnCFDCmCTRLL: <RSCFDn_base> + 0004_H + (10_H × m),
 RSCFDnCFDCmCTRLLH: <RSCFDn_base> + 0005_H + (10_H × m),
 RSCFDnCFDCmCTRHL: <RSCFDn_base> + 0006_H + (10_H × m),
 RSCFDnCFDCmCTRHLH: <RSCFDn_base> + 0007_H + (10_H × m)

Value after reset: 0000 0005_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|------|------|-------|-------|-----------|------|------|----------|-----|-------------|------------|------------|-------|------------|-----|
| | ROM | CRCT | — | — | — | CTMS[1:0] | CTME | ERRD | BOM[1:0] | — | TDCVFI E | SOCOI E | EOCOI E | TAIE | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ALIE | BLIE | OLIE | BORIE | BOEIE | EPIE | EWIE | BEIE | — | — | — | — | RTBO | CSLPR | CHMDC[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 17.96 RSCFDnCFDCmCTR Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 | ROM | Restricted Operation Mode Enable 0: Restricted operation mode is disabled 1: Restricted operation mode is enabled. |
| 30 | CRCT | CRC Error Test Enable 0: The first bit of the reception ID field is not inverted. 1: The first bit of the reception ID field is inverted. |
| 29 to 27 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 26, 25 | CTMS[1:0] | Communication Test Mode Select b ₂₆ b ₂₅ 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode) |
| 24 | CTME | Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled. |
| 23 | ERRD | Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in the RSCFDnCFDCmERFL register are all cleared. 1: Error flags for all error information are displayed. |
| 22, 21 | BOM[1:0] | Bus Off Recovery Mode Select b ₂₂ b ₂₁ 0 0: ISO11898-1 compliant 0 1: Transitions to channel halt mode automatically at bus-off entry 1 0: Transitions to channel halt mode automatically at bus-off end 1 1: Transitions to channel halt mode (in bus-off state) by program request |
| 20 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |

Table 17.96 RSCFDnCFDCmCTR Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 19 | TDCVFIE | Transmitter Delay Compensation Violation Interrupt Enable 0: A transmitter delay compensation violation interrupt is disabled. 1: A transmitter delay compensation violation interrupt is enabled. |
| 18 | SOCOIE | Successful Occurrence Counter Overflow Interrupt Enable 0: A successful occurrence counter overflow interrupt is disabled. 1: A successful occurrence counter overflow interrupt is enabled. |
| 17 | EOCOIE | Error Occurrence Counter Overflow Interrupt Enable 0: An error occurrence counter overflow interrupt is disabled. 1: An error occurrence counter overflow interrupt is enabled. |
| 16 | TAIE | Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled. |
| 15 | ALIE | Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled. |
| 14 | BLIE | Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled. |
| 13 | OLIE | Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled. |
| 12 | BORIE | Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled. |
| 11 | BOEIE | Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled. |
| 10 | EPIE | Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled. |
| 9 | EWIE | Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled. |
| 8 | BEIE | Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled. |
| 7 to 4 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 3 | RTBO | Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0. |
| 2 | CSLPR | Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode |
| 1, 0 | CHMDC[1:0] | Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited |

ROM Bit

When the ROM bit and the CTME bit in the RSCFDnCFDCmCTR register are set to 1, restricted operation mode is enabled. Use the restricted operation mode only when the CTMS[1:0] value in the RSCFDnCFDCmCTR register is 00_B (standard test mode). Modify this bit only in channel halt mode. This bit is always 0 in channel reset mode.

CRCT Bit

This bit is used to test the CRC generation circuit in the RS-CANFD module. Setting this bit to 1 inverts the first bit of the ID field when a message is received. With this inversion of bit, the CRC calculation result does not match the normal CRC value of the received frame, which can detect a CRC error (the CERR bit in the RSCFDnCFDCmERFL register is 1). When using this function, note the following.

- This function is available while the CTME bit in the RSCFDnCFDCmCTR register is 1 (communication test mode enabled).
- This function cannot communicate with other CAN nodes. Use this function for inter-channel communication test (the CmICBCE bit in the RSCFDnCFDGTSTCFG register is 1).
- Bit inversion in the ID field may cause bit stuffing rule violation. In that case, no CRC error is detected but a stuff error is detected.

Modify this bit only in channel halt mode. This bit is set to 0 in channel reset mode.

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCFDnCFDCmERFL register. When this bit is clear to 0, if any error is detected while the flags of bits 14 to 8 in the RSCFDnCFDCmERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CANFD module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CANFD module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CANFD module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCFDnCFDCmCTR register (m = 0 to 2) are set to 10_B and the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RSCFDnCFDCmSTS register are cleared to 00_H.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CANFD module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CANFD module is in the bus off state, the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CANFD module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RS-CANFD module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01_B or at bus off end when the BOM[1:0] bits are 10_B), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TDCVFIE Bit

When the TDCVF flag in the RSCFDnCFDCmFDSTS register is set to 1 after the TDCVFIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

SOCOIE Bit

When the SOCO flag in the RSCFDnCFDCmFDSTS register is set to 1 after the SOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

EOCOIE Bit

When the EOCO flag in the RSCFDnCFDCmFDSTS register is set to 1 after the EOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCFDnCFDCmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCFDnCFDCmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLV flag in the RSCFDnCFDCmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCFDnCFDCmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCFDnCFDCmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCFDnCFDCmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCFDnCFDCmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCFDnCFDCmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly returns from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCFDnCFDCmSTS register to 00_H and also clears the BOSTS flag in the RSCFDnCFDCmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCFDnCFDCmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RSCAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.

Clearing this bit to 0 makes the channel exit channel stop mode.

This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **Section 17.6.2, Channel Modes**. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B. When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits are automatically set to 10_B.

17.4.3.3 RSCFDnCFDCmSTS — Channel m Status Register (m = 0 to 2)

Access: RSCFDnCFDCmSTS register can be read/written in 32-bit units
 RSCFDnCFDCmSTSL, RSCFDnCFDCmSTSH registers can be read/written in 16-bit units
 RSCFDnCFDCmSTSLL, RSCFDnCFDCmSTSLH, RSCFDnCFDCmSTSHL, RSCFDnCFDCmSTSHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmSTS: <RSCFDn_base> + 0008_H + (10_H × m)
 RSCFDnCFDCmSTSL: <RSCFDn_base> + 0008_H + (10_H × m),
 RSCFDnCFDCmSTSH: <RSCFDn_base> + 000A_H + (10_H × m)
 RSCFDnCFDCmSTSLL: <RSCFDn_base> + 0008_H + (10_H × m),
 RSCFDnCFDCmSTSLH: <RSCFDn_base> + 0009_H + (10_H × m),
 RSCFDnCFDCmSTSHL: <RSCFDn_base> + 000A_H + (10_H × m),
 RSCFDnCFDCmSTSHH: <RSCFDn_base> + 000B_H + (10_H × m)

Value after reset: 0000 0005_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|----|----|----|----|----|----|-------|----------|--------|--------|-------|-------|---------|--------|-------|
| | TEC[7:0] | | | | | | | | REC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ESIF | COMSTS | RECSTS | TRMSTS | BOSTS | EPSTS | CSLPSTS | CHLSTS | CRSTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R/W*1 | R | R | R | R | R | R | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.97 RSCFDnCFDCmSTS Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 24 | TEC[7:0] | The transmit error counter (TEC) can be read. |
| 23 to 16 | REC[7:0] | The receive error counter (REC) can be read. |
| 15 to 9 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 8 | ESIF | Error State Indication Flag 0: No CAN FD message whose ESI bit is recessive has been received. 1: At least one CAN FD message whose ESI bit is recessive has been received. |
| 7 | COMSTS | Communication Status Flag 0: Communication is not ready. 1: Communication is ready. |
| 6 | RECSTS | Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception |
| 5 | TRMSTS | Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state |
| 4 | BOSTS | Bus Off Status Flag 0: Not in bus off state 1: In bus off state |
| 3 | EPSTS | Error Passive Status Flag 0: Not in error passive state 1: In error passive state |
| 2 | CSLPSTS | Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode |

Table 17.97 RSCFDnCFDCmSTS Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 1 | CHLTSTS | Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode |
| 0 | CRSTSTS | Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode |

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

ESIF Flag

When the recessive ESI bit is detected in a successfully received message, this flag is set to 1. In loopback mode or mirror mode, the own transmission message is regarded as a received message. To clear this flag to 0, write 0 to this bit by the program. This bit cannot be set to 1 by the program. If the flag setting (to 1) timing matches the writing 0 (by the program) timing, this flag is set to 1.

This flag is set to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag is set to 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state ($TEC[7:0] > 255$) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CANFD module has entered the error passive state ($(128 \leq \text{TEC}[7:0] \leq 255)$ or $(128 \leq \text{REC}[7:0])$), It is cleared to 0 when the RS-CANFD module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

17.4.3.4 RSCFDnCFDCmERFL — Channel m Error Flag Register (m = 0 to 2)

Access: RSCFDnCFDCmERFL register can be read/written in 32-bit units
 RSCFDnCFDCmERFLL, RSCFDnCFDCmERFLH registers can be read/written in 16-bit units
 RSCFDnCFDCmERFLLL, RSCFDnCFDCmERFLLH, RSCFDnCFDCmERFLHL, RSCFDnCFDCmERFLHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmERFL: <RSCFDn_base> + 000C_H + (10_H × m)
 RSCFDnCFDCmERFLL: <RSCFDn_base> + 000C_H + (10_H × m),
 RSCFDnCFDCmERFLH: <RSCFDn_base> + 000E_H + (10_H × m)
 RSCFDnCFDCmERFLLL: <RSCFDn_base> + 000C_H + (10_H × m),
 RSCFDnCFDCmERFLLH: <RSCFDn_base> + 000D_H + (10_H × m),
 RSCFDnCFDCmERFLHL: <RSCFDn_base> + 000E_H + (10_H × m),
 RSCFDnCFDCmERFLHH: <RSCFDn_base> + 000F_H + (10_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CRCREG[14:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | ADERR | BOERR | B1ERR | CERR | AERR | FERR | SERR | ALF | BLF | OVLf | BORF | BOEF | EPF | EWf | BEF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 | R/W*1 |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.98 RSCFDnCFDCmERFL Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 30 to 16 | CRCREG[14:0] | CRC Calculation Data (CRC length:15 bits) A CRC value calculated based on the transmit message or receive message is indicated. |
| 15 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 14 | ADERR | ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected. |
| 13 | BOERR | Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected. |
| 12 | B1ERR | Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected. |
| 11 | CERR | CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected. |
| 10 | AERR | ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected. |
| 9 | FERR | Form Error Flag 0: No form error is detected. 1: Form error is detected. |

Table 17.98 RSCFDnCFDCmERFL Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 8 | SERR | Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected. |
| 7 | ALF | Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected. |
| 6 | BLF | Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected. |
| 5 | OVLf | Overload Flag 0: No overload is detected. 1: Overload is detected. |
| 4 | BORF | Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected. |
| 3 | BOEF | Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected. |
| 2 | EPF | Error Passive Flag 0: No error passive is detected. 1: Error passive is detected. |
| 1 | EWf | Error Warning Flag 0: No error warning is detected. 1: Error warning is detected. |
| 0 | BEF | Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected. |

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is set to 1. Transition to channel reset mode resets these flags to 0.

If the ERRD bit in the RSCFDnCFDCmCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCFDnCFDCmERFL is detected, the flag bits are only set to 1 by the error event if bits 14 to 8 were all 0 at the time when the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCFDnCFDCmCTR register is set to 1 (communication test mode is enabled), if transmit or receive message is a classical CAN frame (CRC length = 15 bits), this flag is updated and the CRC value calculated based on the message can be read. When a CAN FD frame is sent or received, the value of CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register is updated. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0. This bit is always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected even though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected even though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of a bus lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVL F Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCFDnCFDCmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCFDnCFDCmCTR register is set to 1 (forcible return from the bus off state).
- The BOM[1:0] bits in the RSCFDnCFDCmCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCFDnCFDCmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is entered (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is entered when the BOM[1:0] bits in the RSCFDnCFDCmCTR register (m = 0 to 2) are set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag is set to 1 when the error passive state is entered ($(128 \leq \text{TEC}[7:0] \leq 255)$ or $(128 \leq \text{REC}[7:0])$).

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWf Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, BOERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCFDnCFDCmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write "0" to the given flag and "1" to the other flags.

17.4.3.5 RSCFDnCFDCmDCFG — Channel m Data Bit Rate Configuration Register (m = 0 to 2)

Access: RSCFDnCFDCmDCFG register can be read/written in 32-bit units
 RSCFDnCFDCmDCFGL, RSCFDnCFDCmDCFGLH registers can be read/written in 16-bit units
 RSCFDnCFDCmDCFGLL, RSCFDnCFDCmDCFGLH, RSCFDnCFDCmDCFGLH, RSCFDnCFDCmDCFGLH,
 RSCFDnCFDCmDCFGLH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmDCFG: <RSCFDn_base> + 0500_H + (20_H × m)
 RSCFDnCFDCmDCFGL: <RSCFDn_base> + 0500_H + (20_H × m),
 RSCFDnCFDCmDCFGLH: <RSCFDn_base> + 0502_H + (20_H × m)
 RSCFDnCFDCmDCFGLL: <RSCFDn_base> + 0500_H + (20_H × m),
 RSCFDnCFDCmDCFGLH: <RSCFDn_base> + 0501_H + (20_H × m),
 RSCFDnCFDCmDCFGLH: <RSCFDn_base> + 0502_H + (20_H × m),
 RSCFDnCFDCmDCFGLH: <RSCFDn_base> + 0503_H + (20_H × m)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|-----------|-----|-----|-----------|-------------|-----|-----|-------------|-----|-----|-----|
| | — | — | — | — | — | DSJW[2:0] | | | — | DTSEG2[2:0] | | | DTSEG1[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | DBRP[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.99 RSCFDnCFDCmDCFG Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 27 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 26 to 24 | DSJW[2:0] | Data Bit Rate Resynchronization Jump Width Control b26 b25 b24 0 0 0: 1 Tq 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq |
| 23 | Reserved | This bit is read as the value after reset. The write value should be the value after reset. |
| 22 to 20 | DTSEG2[2:0] | Data Bit Rate Time Segment 2 Control b22 b21 b20 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq |

Table 17.99 RSCFDnCFDCmDCFG Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 19 to 16 | DTSEG1[3:0] | Data Bit Rate Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: 2 Tq 0 0 1 0: 3 Tq 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq |
| 15 to 8 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 7 to 0 | DBRP[7:0] | Data Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 255), the data bit rate prescaler divides fCAN by (P + 1). |

Modify the RSCFDnCFDCmDCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. When only classical CAN frames are used in CAN FD mode, set the RSCFDnCFDCmDCFG register to the value equal to the set RSCFDnCFDCmNCFG register value. For the description and settings of bit timing parameters, see **Section 17.11.1, Initial Settings**.

DSJW[2:0] Bits

These bits specify the resynchronization jump width of data bit rate as a Tq value. Possible values are 1 to 8 Tq. Specify a value equal to or smaller than the DTSEG2[2:0] bits value.

DTSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of data bit rate.

Possible values are 2 to 8 Tq.

Specify a value equal to or smaller than the DTSEG1[3:0] bits value.

DTSEG1[3:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of data bit rate as a Tq value.

Possible values are 2 to 16 Tq.

DBRP[7:0] Bits

The clock obtained by dividing the CAN clock (f_{CAN}) by the data bit rate prescaler ($(DBRP[7:0] + 1)$) becomes $CANmTq(D)$ clock ($f_{CANTQ(D)m}$). One clock of the $CANmTq(D)$ clock becomes one Time Quantum (Tq).

Be sure to specify the same value for both $NBRP[9:0]$ and $DBRP[7:0]$.

To specify different values for the nominal bit rate and the data bit rate, change the values of the $RCFGDnCFGmNCFG.NTSEG1$ and $NTSEG2$ bits and $RSCFDnCFDCmDCFG.DTSEG1$ and $DTSEG2$ bits, respectively.

When the $TDCE$ bit is set to 1 (Transmitter delay compensation is enabled) in the $RSCFDnCFDCmFDCFG$ register, set the equal value of 1 or less to the bits $NBRP[9:0]$ and $DBRP[7:0]$.

17.4.3.6 RSCFDnCFDCmFDCFG — Channel m CAN FD Configuration Register (m = 0 to 2)

Access: RSCFDnCFDCmFDCFG register can be read/written in 32-bit units
 RSCFDnCFDCmFDCFGL, RSCFDnCFDCmFDCFGH registers can be read/written in 16-bit units
 RSCFDnCFDCmFDCFGLL, RSCFDnCFDCmFDCFGLH, RSCFDnCFDCmFDCFGHL,
 RSCFDnCFDCmFDCFGHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmFDCFG: <RSCFDn_base> + 0504_H + (20_H × m)
 RSCFDnCFDCmFDCFGL: <RSCFDn_base> + 0504_H + (20_H × m),
 RSCFDnCFDCmFDCFGH: <RSCFDn_base> + 0506_H + (20_H × m)
 RSCFDnCFDCmFDCFGLL: <RSCFDn_base> + 0504_H + (20_H × m),
 RSCFDnCFDCmFDCFGLH: <RSCFDn_base> + 0505_H + (20_H × m),
 RSCFDnCFDCmFDCFGHL: <RSCFDn_base> + 0506_H + (20_H × m),
 RSCFDnCFDCmFDCFGHH: <RSCFDn_base> + 0507_H + (20_H × m)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|------|------|------|-----------|-------|-------|----|-----------|-----|-----|-----|-------------|-----|-----|
| | — | — | REFE | FDOE | TMME | GWBR S | GWDFD | GWEN | — | TDCO[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | ESIC | TDCE | TDCOC | — | — | — | — | — | EOCCFG[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 17.100 RSCFDnCFDCmFDCFG Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31, 30 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 29 | REFE | Reception data edge filter enable bit 0: Reception data edge filter is disabled. 1: Reception data edge filter is enabled. |
| 28 | FDOE | FD-only mode enable bit 0: FD-only mode is disabled. 1: FD-only mode is enabled. |
| 27 | TMME | Transmit Buffer Merge Mode Enable 0: Transmit buffer merge mode is disabled. 1: Transmit buffer merge mode is enabled. |
| 26 | GWBR S | Gateway BRS 0: A frame is transmitted with the BRS bit in the received frame set to 0. 1: A frame is transmitted with the BRS bit in the received frame set to 1. |
| 25 | GWDFD | Gateway FDF 0: A frame is transmitted regarding the received frame as a classical CAN frame. 1: A frame is transmitted regarding the received frame as a CAN FD frame. |
| 24 | GWEN | CAN-CAN FD Gateway Enable 0: The CAN-CAN FD gateway is disabled. 1: The CAN-CAN FD gateway is enabled. |
| 23 | Reserved | This bit is read as the value after reset. The write value should be the value after reset. |
| 22 to 16 | TDCO[6:0] | Transmitter Delay Compensation Offset These bits are set to the transmitter delay compensation offset value. |
| 15 to 11 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |

Table 17.100 RSCFDnCFDCmFDCFG Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 10 | ESIC | Error State Display Mode Select 0: Always displays the node error state. 1: When the node is not in the error passive state: Displays the message buffer error state. When the node is in the error passive state: Displays the node error state. |
| 9 | TDCE | Transmitter Delay Compensation Enable 0: Transmitter delay compensation is disabled. 1: Transmitter delay compensation is enabled. |
| 8 | TDCOC | Transmitter Delay Compensation Measurement Select 0: Measurement and offset 1: Only offset |
| 7 to 3 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 2 to 0 | EOCCFG[2:0] | Error Occurrence Counting Method Select b2 b1 b0 0 0 0: All transmit messages and receive messages 0 0 1: All transmit messages 0 1 0: All receive messages 0 1 1: Setting prohibited 1 0 0: Only data phase of transmitted or received CAN FD message 1 0 1: Only data phase of transmitted CAN FD message 1 1 0: Only data phase of received CAN FD message 1 1 1: Setting prohibited |

REFE bit

Setting this bit to 1 enables reception data edge filtering when the idle condition is detected, and a dominant level with less than 2 time quanta is ignored. A dominant level with more than or equal to 2 time quanta is detected as an edge. Modify this bit only in channel reset mode.

FDOE bit

Setting this bit to 1 enables FD-only mode. When data is transmitted, a CAN FD frame will be sent regardless of the settings to the CFFDF bit in the RSCFDnCFDCFFDCSTSk register or the TMFDF bit in the RSCFDnCFDTMFDCSTRp register. When a Classical CAN frame is received, a form error is detected. Modify this bit only in channel reset mode.

TMME Bit

Setting this bit to 1 enables transmit buffer merge mode. Modify this bit only in channel reset mode or channel halt mode.

GWBRs Bit

When the GWEN bit is 1, the BRS bit in a CAN FD frame to be transmitted by the gateway function is set. When the GWFDF bit is set to 0, write 0 to this bit. Modify this bit only in channel reset mode.

GWFDF Bit

When the GWEN bit is 1, the FDF bit in a CAN FD frame to be transmitted by the gateway function is set. Modify this bit only in channel reset mode.

GWEN Bit

This bit is used to control the operation of the transmit/receive FIFO buffer with the CFM[1:0] bits in the RSCFDnCFDCFCCK register set to 10_B (gateway mode).

Setting this bit to 1 enables the CAN-CAN FD gateway, enabling transmission in a format different from that of frames received by the gateway function. Received frames are replaced in accordance with the settings of the GWDFD bit and the GWBRS bit. When the DLC value in the received classical CAN frame is 1001_B or more and the GWDFD bit is set to 1 (CAN FD frame), the DLC value is replaced with 1000_B.

While this bit is set to 1, do not perform routing the following frames by using the gateway function.

- CAN FD frames with a payload length of more than 8 bytes
- Remote frames

While this bit is set to 1, the following frame should be transmitted from the channel according to the setting of GWDFD.

- When GWDFD bit is set to 0, only classical CAN frame should be transmitted.
- When GWDFD bit is set to 1, only CAN FD frame should be transmitted.

Modify this bit only in channel reset mode.

Table 17.101 shows the settings and formats of transmit frame and receive frame while the CAN-CAN FD gateway is enabled.

Table 17.101 Operation when the CAN-CAN FD Gateway is Enabled

| Receive Frame | | | GWDFD Bit | Transmit Frame | | |
|---------------|-----------|-------------------------|-----------|----------------|--------------------------------|---------------------------------|
| Format | BRS Bit | Received DLC Value | | Format | BRS Bit | DLC Value to be Transmitted |
| Classical CAN | None | DLC ≤ 1000 _B | 0 | Classical CAN | None | Not replaced |
| | | DLC > 1000 _B | | | | |
| CAN FD | Arbitrary | DLC ≤ 1000 _B | | | | |
| Classical CAN | None | DLC ≤ 1000 _B | 1 | CAN FD | According to GWBRS bit setting | Not replaced |
| | | DLC > 1000 _B | | | | Replaced with 1000 _B |
| CAN FD | Arbitrary | DLC ≤ 1000 _B | | | | Not replaced |

TDCO[6:0] Bits

These bits set the SSP offset value. How to use this value depends on the TDCOC bit in the RSCFDnCFDCmFDCFG register. These bits are based on CAN clock frequency (fCAN).

When the TDCOC bit is set to 0, the transmitter delay compensation result equals to the total value of the measured delay value and the TDCO[6:0] value (rounded down to the nearest integer Tq).

When the TDCOC bit is set to 1, the transmitter delay compensation result equals to the TDCO[6:0] value.

The SSP offset value = (set value of TDCO[6:0] bits + 1).

Modify these bits only in channel reset mode or channel halt mode.

ESIC Bit

When the ESIC bit is set to 1, if the channel is in the error active state, the ESI bit value (CFESI bit in the RSCFDnCFDCFFDCSTSk register or TMESI bit in the RSCFDnCFDTMFDCTRp register) set in the transmit/receive FIFO buffer or transmit buffer is transmitted as an ESI bit value of the transmit message. When the channel is in the error passive state or the ESIC bit is set to 0, the channel status is transmitted as an ESI bit value. Modify this bit only in channel reset mode or channel halt mode.

Table 17.102 ESI Value to Be Transmitted

| ESIC Bit | Channel Status | ESI Value to be Transmitted |
|----------|----------------|--|
| 0 | Error active | 0 (error active node) |
| | Error passive | 1 (error passive node) |
| 1 | Error active | ESI value set in the transmit/receive FIFO buffer or transmit buffer (CFESI bit in the RSCFDnCFDCFFDCSTSk register or TMESI bit in the RSCFDnCFDTMFDCTRp register) |
| | Error passive | 1 (error passive node) |

TDCE Bit

Setting this bit to 1 enables transmitter delay compensation. Modify this bit only in channel reset mode or channel halt mode.

TDCOC Bit

When this bit is set to 0, the SSP position is defined by the total of the measured delay value and the SSP offset value (fixed value).

When this bit is set to 1, the SSP position is defined only by the SSP offset value.

Modify this bit only in channel reset mode or channel halt mode.

EOCCFG[2:0] Bits

These bits are used to select a frame format and a transmission/reception direction when the error occurrence counter counts CAN bus errors.

Modify these bits only in channel reset mode or channel halt mode.

17.4.3.7 RSCFDnCFDCmFDCTR — Channel m CAN FD Control Register (m = 0 to 2)

Access: RSCFDnCFDCmFDCTR register can be read/written in 32-bit units
 RSCFDnCFDCmFDCTRL, RSCFDnCFDCmFDCTRH registers can be read/written in 16-bit units
 RSCFDnCFDCmFDCTRL, RSCFDnCFDCmFDCTRLH, RSCFDnCFDCmFDCTRHL,
 RSCFDnCFDCmFDCTRHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmFDCTR: <RSCFDn_base> + 0508_H + (20_H × m)
 RSCFDnCFDCmFDCTRL: <RSCFDn_base> + 0508_H + (20_H × m),
 RSCFDnCFDCmFDCTRH: <RSCFDn_base> + 050A_H + (20_H × m)
 RSCFDnCFDCmFDCTRL: <RSCFDn_base> + 0508_H + (20_H × m),
 RSCFDnCFDCmFDCTRLH: <RSCFDn_base> + 0509_H + (20_H × m),
 RSCFDnCFDCmFDCTRHL: <RSCFDn_base> + 050A_H + (20_H × m),
 RSCFDnCFDCmFDCTRHH: <RSCFDn_base> + 050B_H + (20_H × m)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SOCCLR R | EOCCLR R |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 17.103 RSCFDnCFDCmFDCTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 1 | SOCCLR | Successful Occurrence Counter Clear Setting the SOCCLR bit to 1 clears the successful occurrence counter. This bit is always read as 0. |
| 0 | EOCCLR | Error Occurrence Counter Clear Setting the EOCCLR bit to 1 clears the error occurrence counter. This bit is always read as 0. |

SOCCLR Bit

Setting this bit to 1 clears the successful occurrence counter (SOC[7:0] bits in the RSCFDnCFDCmFDSTS register). This bit is automatically cleared to 0.

EOCCLR Bit

Setting this bit to 1 clears the error occurrence counter (EOC[7:0] bits in the RSCFDnCFDCmFDSTS register). This bit is automatically cleared to 0.

17.4.3.8 RSCFDnCFDCmFDSTS — Channel m CAN FD Status Register (m = 0 to 2)

Access: RSCFDnCFDCmFDSTS register can be read/written in 32-bit units
 RSCFDnCFDCmFDSTSL, RSCFDnCFDCmFDSTSH registers can be read/written in 16-bit units
 RSCFDnCFDCmFDSTSL, RSCFDnCFDCmFDSTSLH, RSCFDnCFDCmFDSTSHL,
 RSCFDnCFDCmFDSTSHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmFDSTS: <RSCFDn_base> + 050C_H + (20_H × m)
 RSCFDnCFDCmFDSTSL: <RSCFDn_base> + 050C_H + (20_H × m),
 RSCFDnCFDCmFDSTSH: <RSCFDn_base> + 050E_H + (20_H × m)
 RSCFDnCFDCmFDSTSL: <RSCFDn_base> + 050C_H + (20_H × m),
 RSCFDnCFDCmFDSTSLH: <RSCFDn_base> + 050D_H + (20_H × m),
 RSCFDnCFDCmFDSTSHL: <RSCFDn_base> + 050E_H + (20_H × m),
 RSCFDnCFDCmFDSTSHH: <RSCFDn_base> + 050F_H + (20_H × m)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|----|----|----|----|----|-------|-------|----------|-----------|----|----|----|----|----|----|
| | SOC[7:0] | | | | | | | | EOC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | SOCO | EOCO | TDCVF | TDCR[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W*1 | R/W*1 | R/W*1 | R | R | R | R | R | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.104 RSCFDnCFDCmFDSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 24 | SOC[7:0] | Successful Occurrence Counter The successful occurrence counter value can be read. |
| 23 to 16 | EOC[7:0] | Error Occurrence Counter The error occurrence counter value can be read. |
| 15 to 10 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 9 | SOCO | Successful Occurrence Counter Overflow Flag 0: The successful occurrence counter does not overflow. 1: The successful occurrence counter has overflowed. |
| 8 | EOCO | Error Occurrence Counter Overflow Flag 0: The error occurrence counter does not overflow. 1: The error occurrence counter has overflowed. |
| 7 | TDCVF | Transmitter Delay Compensation Violation Flag 0: No transmitter delay compensation violation is present. 1: A transmitter delay compensation violation is present. |
| 6 to 0 | TDCR[6:0] | Transmitter Delay Compensation Result Status The transmitter delay compensation result can be read. |

SOC[7:0] Bits

These bits show the successful occurrence counter value. The successful occurrence counter is incremented upon completion of message reception or transmission without an error. This counter stops counting when it reaches FF_H . In loopback mode, this counter is incremented twice.

These bits are cleared to 0 by writing 1 to the SOCCLR bit in the RSCFDnCFDCmCTR register. These bits are set to 0 in channel reset mode.

EOC[7:0] Bits

These bits show the error occurrence counter value. The error occurrence counter is incremented each time an error occurs according to the condition specified by the EOCCFG[2:0] bits in the RSCFDnCFDCmFDCFG register. This counter stops counting when it reaches FF_H .

These bits are cleared to 0 by writing 1 to the EOCCLR bit in the RSCFDnCFDCmCTR register. These bits are set to 0 in channel reset mode.

SOCO Flag

This bit indicates that successful occurrence counter overflow has occurred.

This flag is set to 1 when message reception or transmission is completed while the SOC[7:0] value has reached FF_H . This flag is set to 0 in channel reset mode.

EOCO Flag

This bit indicates that error occurrence counter overflow has occurred.

This flag is set to 1 when a CAN bus error is detected under the condition specified by the EOCCFG[2:0] bits in the RSCFDnCFDCmFDCFG register when the EOC[7:0] value has reached FF_H . This flag is set to 0 in channel reset mode.

TDCVF Flag

This bit indicates violation of transmitter delay compensation.

The transmit data is compared with the reception CAN bus level delayed due to the transceiver's loop delay. This delay changes due to physical factors such as temperature. Because the TDCR[6:0] flags are updated for each message, temporary maximum delay cannot be confirmed.

This bit is set to 1 when the transmitter delay compensation exceeds the maximum compensation $3 \text{ CANm bit times} - 2 \text{ fCAN}$ (CANm bit time is the value of data bit rate).

This flag is set to 0 in channel reset mode.

TDCR[6:0] Flags

These bits indicate the transmitter delay compensation result as a multiple of CAN clock frequency (fCAN).

This result depends on the settings of the TDCOC bit and TDCO[6:0] bits in the RSCFDnCFDCmFDCFG register.

These flags are updated at a falling edge between the FDF bit and res bit when the TDCE bit in the RSCFDnCFDCmFDCFG register is set to 1 (transmitter delay compensation enable) and also the TDCOC bit in the RSCFDnCFDCmFDCFG register is set to 0 (measurement and offset).

This flag is set to 0 in channel reset mode.

NOTE

To clear the flag of this register to 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

17.4.3.9 RSCFDnCFDCmFDCRC — Channel m CAN FD CRC Register (m = 0 to 2)

Access: RSCFDnCFDCmFDCRC register can be read only in 32-bit units
 RSCFDnCFDCmFDCRCL, RSCFDnCFDCmFDCRCH registers can be read only in 16-bit units
 RSCFDnCFDCmFDCRCLL, RSCFDnCFDCmFDCRCLH, RSCFDnCFDCmFDCRCHL,
 RSCFDnCFDCmFDCRCHH registers can be read only in 8-bit units

Address: RSCFDnCFDCmFDCRC: <RSCFDn_base> + 0510_H + (20_H × m)
 RSCFDnCFDCmFDCRCL: <RSCFDn_base> + 0510_H + (20_H × m),
 RSCFDnCFDCmFDCRCH: <RSCFDn_base> + 0512_H + (20_H × m)
 RSCFDnCFDCmFDCRCLL: <RSCFDn_base> + 0510_H + (20_H × m),
 RSCFDnCFDCmFDCRCLH: <RSCFDn_base> + 0511_H + (20_H × m),
 RSCFDnCFDCmFDCRCHL: <RSCFDn_base> + 0512_H + (20_H × m),
 RSCFDnCFDCmFDCRCHH: <RSCFDn_base> + 0513_H + (20_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|----|----|----|----|----|-----------|----|----|----|----|----|---------------|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | — | | | SCNT[3:0] | | | — | | | CRCREG[20:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CRCREG[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.105 RSCFDnCFDCmFDCRC Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 28 | Reserved | These bits are read as the value after reset. |
| 27 to 24 | SCNT[3:0] | Stuff count bit Indicate a value of the stuff count in a CAN FD frame. Bits 25 to 27 indicates the Gray-coded value of the stuff bit count modulo 8 in the transmitted/received frames. Bit 24 indicates an even parity value of bits 25 to 27. |
| 23 to 21 | Reserved | These bits are read as the value after reset. |
| 20 to 0 | CRCREG[20:0] | CRC Calculation Data (CRC Length:17 Bit or 21 Bit) These bits show the CRC value calculated based on the transmit message or receive message. When the CRC length is 17 bits, bits b20 to b17 are read as 0. |

SCNT[3:0] Flags

When the CTME bit in the RSCFDnCFDCmCTR register is set to 1 (communication test mode enabled), a stuff count bit value of the CAN FD frame can be read if a message transmitted/received is a CAN FD frame. When the CTME bit is 0 (communication test mode disabled), this flag is always read as 0. These flags are updated at the first bit in the CRC field of the CAN FD frame. These bits are cleared to 0 in channel reset mode.

CRCREG[20:0] Flags

When the CTME bit in the RSCFDnCFDCmCTR register is 1 (communication test mode enabled), if transmit or receive message is a CAN FD frame (CRC length = 17 or 21 bits), these flags are updated and the CRC value calculated based on the message can be read. When the CRC length of the message is 17 bits, bits b20 to b17 are always read as 0. When a classical CAN frame is transmitted or received, the CRCREG[14:0] value in the RSCFDnCFDCmERFL register is updated. When the CTME bit is 0 (communication test mode disabled), these bits are always read as 0.

17.4.4 Details of Global-Related Registers

17.4.4.1 RSCFDnCFDGCFCG — Global Configuration Register

Access: RSCFDnCFDGCFCG register can be read/written in 32-bit units
 RSCFDnCFDGCFCGL, RSCFDnCFDGCFCGH registers can be read/written in 16-bit units
 RSCFDnCFDGCFCGLL, RSCFDnCFDGCFCGLH, RSCFDnCFDGCFCGHL, RSCFDnCFDGCFCGHH registers
 can be read/written in 8-bit units

Address: RSCFDnCFDGCFCG: <RSCFDn_base> + 0084_H
 RSCFDnCFDGCFCGL: <RSCFDn_base> + 0084_H, RSCFDnCFDGCFCGH: <RSCFDn_base> + 0086_H
 RSCFDnCFDGCFCGLL: <RSCFDn_base> + 0084_H, RSCFDnCFDGCFCGLH: <RSCFDn_base> + 0085_H,
 RSCFDnCFDGCFCGHL: <RSCFDn_base> + 0086_H, RSCFDnCFDGCFCGHH: <RSCFDn_base> + 0087_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|-----|------|----------|-----|-----|-----|-----|-----------|-----|-----|-----|-----|------|-----|-----|
| | ITRCP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSBTCS[2:0] | | TSSS | TSP[3:0] | | | — | — | CMPO C | DCS | MME | DRE | DCE | TPRI | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.106 RSCFDnCFDGCFCG Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 16 | ITRCP[15:0] | Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000 _H is prohibited when the interval timer is in use. |
| 15 to 13 | TSBTCS[2:0] | Timestamp Clock Source Select b ¹⁵ b ¹⁴ b ¹³ 0 0 0: Channel 0 nominal bit time clock 0 0 1: Channel 1 nominal bit time clock 0 1 0: Channel 2 nominal bit time clock 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited |
| 12 | TSSS | Timestamp Source Select 0: pclk/2 ¹ 1: Nominal bit time clock |

Table 17.106 RSCFDnCFDGCFG Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 11 to 8 | TSP[3:0] | Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768 |
| 7, 6 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 5 | CMPOC | Payload Overflow Mode Select 0: No message is stored. 1: Messages are stored and payloads exceeding the buffer size are discarded. |
| 4 | DCS | CAN Clock Source Select*2 0: clkc 1: clk_xincan |
| 3 | MME | Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled. |
| 2 | DRE | DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled. |
| 1 | DCE | DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled. |
| 0 | TPRI | Transmit Priority Select 0: ID priority 1: Transmit buffer number priority |

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Note 2. For the CAN clock frequency settings, see **Table 17.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/P1M-E.**

Modify the RSCFDnCFDGCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See **Section 17.8.3.1, Interval Transmission Function.**

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the nominal bit time clock that will be the clock source of the timestamp counter. However, do not select the channel that handles the CAN FD frames.

TSSS Bit

This bit is used to select a clock source of the timestamp counter. Select pclk if there is no channel that handles only classical CAN frames.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

CMPOC Bit

This bit is used to select operation in case the payload length of received message exceeds the payload storage size of the storage buffer.

When this bit is 0, the received message in which the payload overflows is not stored in the buffer.

When this bit is 1, the received message in which the payload overflows is stored in the buffer, and depending on the DRE bit the received DLC value or the DLC value of the receive rule is stored in the buffer. At this time, payloads exceeding the buffer's payload storage size are discarded.

The buffer's payload storage size is set by the following bits.

- Receive buffer: RMPLS[1:0] bits in the RSCFDnCFDRMNB register
- Receive FIFO buffer: RFPLS[2:0] bits in the RSCFDnCFDRFCCx register
- Transmit/receive FIFO buffer: CFPLS[2:0] bits in the RSCFDnCFDCFCCk register

DCS Bit

When this bit is set to 0, clk is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN).

For the CAN clock frequency settings, see **Table 17.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/P1M-E**.

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte that exceeds the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCFDnCFDGAFLP0_j register to 0000_B before clearing the DCE bit in the RSCFDnCFDGCFCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the transmit buffer with the smallest number has the highest priority.

When using the transmit queue, this bit should be set to 0.

17.4.4.2 RSCFDnCFDGCTR — Global Control Register

Access: RSCFDnCFDGCTR register can be read/written in 32-bit units
 RSCFDnCFDGCTRL, RSCFDnCFDGCTRH registers can be read/written in 16-bit units
 RSCFDnCFDGCTRLL, RSCFDnCFDGCTRLLH, RSCFDnCFDGCTRHL, RSCFDnCFDGCTRHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGCTR: <RSCFDn_base> + 0088_H
 RSCFDnCFDGCTRL: <RSCFDn_base> + 0088_H, RSCFDnCFDGCTRH: <RSCFDn_base> + 008A_H
 RSCFDnCFDGCTRLL: <RSCFDn_base> + 0088_H, RSCFDnCFDGCTRLLH: <RSCFDn_base> + 0089_H,
 RSCFDnCFDGCTRHL: <RSCFDn_base> + 008A_H, RSCFDnCFDGCTRHH: <RSCFDn_base> + 008B_H

Value after reset: 0000 0005_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|-------------|--------|------|------|----|----|----|----|----|-------|-----------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSRST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | CMPOF IE | THLEIE | MEIE | DEIE | — | — | — | — | — | GSLPR | GMDC[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 17.107 RSCFDnCFDGCTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 17 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 16 | TSRST | Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0. |
| 15 to 12 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 11 | CMPOFIE | Payload Overflow Interrupt Enable 0: A payload overflow interrupt is disabled. 1: A payload overflow interrupt is enabled. |
| 10 | THLEIE | Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled. |
| 9 | MEIE | FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled. |
| 8 | DEIE | DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled. |
| 7 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | GSLPR | Global Stop Mode 0: Other than global stop mode 1: Global stop mode |
| 1, 0 | GMDC[1:0] | Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited |

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCFDnCFDGTSC register is cleared to 0000_H.

CMPOFIE Bit

When the CMPOF flag in the RSCFDnCFDGERFL register is set to 1 after the CMPOFIE bit is set to 1, an interrupt request occurs. Modify this bit only in global reset mode.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RS-CANFD module into global stop mode.
Clearing this bit to 0 releases the RS-CANFD module from global stop mode.
This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CANFD module (global operating mode, global reset mode, or global test mode). For details, see **Section 17.6.1, Global Modes**. Setting the GSLPR bit to 1 when in global reset mode places the RS-CANFD module in global stop mode.

17.4.4.3 RSCFDnCFDGSTS — Global Status Register

Access: RSCFDnCFDGSTS register can be read only in 32-bit units
RSCFDnCFDGSTSL, RSCFDnCFDGSTSH registers can be read only in 16-bit units
RSCFDnCFDGSTSL, RSCFDnCFDGSTSLH, RSCFDnCFDGSTSHL, RSCFDnCFDGSTSHH registers can be read only in 8-bit units

Address: RSCFDnCFDGSTS: <RSCFDn_base> + 008C_H
RSCFDnCFDGSTSL: <RSCFDn_base> + 008C_H, RSCFDnCFDGSTSH: <RSCFDn_base> + 008E_H
RSCFDnCFDGSTSL: <RSCFDn_base> + 008C_H, RSCFDnCFDGSTSLH: <RSCFDn_base> + 008D_H,
RSCFDnCFDGSTSHL: <RSCFDn_base> + 008E_H, RSCFDnCFDGSTSHH: <RSCFDn_base> + 008F_H

Value after reset: 0000 000D_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----------|---------|---------|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | GRAMINIT | GSLPSTS | GHLTSTS | GRSTSTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.108 RSCFDnCFDGSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 4 | Reserved | When read, the value after reset is returned. |
| 3 | GRAMINIT | CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing. |
| 2 | GSLPSTS | Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode |
| 1 | GHLTSTS | Global Test Status Flag 0: Not in global test mode 1: In global test mode |
| 0 | GRSTSTS | Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode |

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has transitioned to a mode other than global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

17.4.4.4 RSCFDnCFDGERFL — Global Error Flag Register

Access: RSCFDnCFDGERFL register can be read/written in 32-bit units
 RSCFDnCFDGERFLL, RSCFDnCFDGERFLH registers can be read/written in 16-bit units
 RSCFDnCFDGERFLLL, RSCFDnCFDGERFLLH, RSCFDnCFDGERFLHL, RSCFDnCFDGERFLHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGERFL: <RSCFDn_base> + 0090_H
 RSCFDnCFDGERFLL: <RSCFDn_base> + 0090_H, RSCFDnCFDGERFLH: <RSCFDn_base> + 0092_H
 RSCFDnCFDGERFLLL: <RSCFDn_base> + 0090_H, RSCFDnCFDGERFLLH: <RSCFDn_base> + 0091_H,
 RSCFDnCFDGERFLHL: <RSCFDn_base> + 0092_H, RSCFDnCFDGERFLHH: <RSCFDn_base> + 0093_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | EEF2 | EEF1 | EEF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R/W*1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|-------|-------|-----|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | CMPOF | THLES | MES | DEF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R | R | R/W*1 |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.109 RSCFDnCFDGERFL Register Contents

| Bit Position | Bit Name | Function |
|---------------------------|----------|---|
| 31 to 19, 15, 14, 7, 6, 4 | Reserved | When read, the value after reset is returned. When writing these bits, write the value after reset. |
| 18 | EEF2 | ECC Error Flag for Channel 2 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority |
| 17 | EEF1 | ECC Error Flag for Channel 1 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority |
| 16 | EEF0 | ECC Error Flag for Channel 0 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority |
| 13 to 8, 5 | Reserved | When read, the undefined value is returned. When writing these bits, write the value after reset. |
| 3 | CMPOF | Payload Overflow Flag 0: No payload overflow has occurred. 1: A payload overflow has occurred. |
| 2 | THLES | Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred. |
| 1 | MES | FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred. |
| 0 | DEF | DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred. |

All flags in the RSCFDnCFDGERFL register are cleared to 0 in global reset mode.

EEFm Flag

When a 2-bit ECC error is detected during the transmission priority determination of channel m (m = 0 to 2), the EEFm flag is set to 1, disabling message transmission. This flag can be cleared to 0 by writing 0 by the program.

CMPOF Flag

When a payload overflow occurs in any of channel m (m = 0 to 2), the CMPOF flag is set to 1. This flag can be cleared to 0 by writing 0 to this bit by the program.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCFDnCFDTHLSTSm register (m = 0 to 5) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCFDnCFDRFSTSx register (x = 0 to 7) or the CFMLT flags in the RSCFDnCFDCFSTSk register (k = 0 to 8) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

NOTE

To clear the flag of this register to 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

17.4.4.5 RSCFDnCFDGTSC — Global Timestamp Counter Register

Access: RSCFDnCFDGTSC register can be read only in 32-bit units.
RSCFDnCFDGTSC_L, RSCFDnCFDGTSC_H registers can be read only in 16-bit units.

Address: RSCFDnCFDGTSC: <RSCFDn_base> + 0094_H
RSCFDnCFDGTSC_L: <RSCFDn_base> + 0094_H, RSCFDnCFDGTSC_H: <RSCFDn_base> + 0096_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.110 RSCFDnCFDGTSC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | Reserved | These bits are read as the value after reset. |
| 15 to 0 | TS[15:0] | Timestamp Value The timestamp counter value can be read. Counter Value: 0000 _H to FFFF _H |

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer. Furthermore, the TS[15:0] value is stored in the transmit history buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RSCFDnCFDGCFCFG register is 0 (pclk):
The timestamp counter starts counting when the RSCAN module has transitioned to global operating mode.
This counter stops counting when the RSCAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm nominal bit time clock):
The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

17.4.4.6 RSCFDnCFDGTINTSTS0 — Global TX Interrupt Status Register 0

Access: RSCFDnCFDGTINTSTS0 register can be read only in 32-bit units
 RSCFDnCFDGTINTSTS0L, RSCFDnCFDGTINTSTS0H registers can be read only in 16-bit units
 RSCFDnCFDGTINTSTS0LL, RSCFDnCFDGTINTSTS0LH, RSCFDnCFDGTINTSTS0HL,
 RSCFDnCFDGTINTSTS0HH registers can be read only in 8-bit units

Address: RSCFDnCFDGTINTSTS0: <RSCFDn_base> + 0460_H
 RSCFDnCFDGTINTSTS0L: <RSCFDn_base> + 0460_H, RSCFDnCFDGTINTSTS0H: <RSCFDn_base> +
 0462_H
 RSCFDnCFDGTINTSTS0LL: <RSCFDn_base> + 0460_H, RSCFDnCFDGTINTSTS0LH: <RSCFDn_base> +
 0461_H,
 RSCFDnCFDGTINTSTS0HL: <RSCFDn_base> + 0462_H, RSCFDnCFDGTINTSTS0HH: <RSCFDn_base> +
 0463_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|-----------------|-----------------|-----------------|-----------------|-----------------|----|----|----|-----------------|-----------------|-----------------|-----------------|-----------------|
| | — | — | — | — | — | — | — | — | — | — | — | THIF2 | CFTIF2 | TQIF2 | TAIF2 | TSIF2 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | THIF1 | CFTIF1 | TQIF1 | TAIF1 | TSIF1 | — | — | — | THIF0 | CFTIF0 | TQIF0 | TAIF0 | TSIF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R | R | R | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} | R ^{*1} |

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 17.111 RSCFDnCFDGTINTSTS0 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 21 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 20 | THIF2 | Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested. |
| 19 | CFTIF2 | Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested. |
| 18 | TQIF2 | Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested. |
| 17 | TAIF2 | Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested. |
| 16 | TSIF2 | Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested. |
| 15 to 13 | Reserved | These bits are read as the value after reset. |
| 12 | THIF1 | Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested. |
| 11 | CFTIF1 | Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested. |
| 10 | TQIF1 | Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested. |

Table 17.111 RSCFDnCFDGTINTSTS0 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 9 | TAIF1 | Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested. |
| 8 | TSIF1 | Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested. |
| 7 to 5 | Reserved | These bits are read as the value after reset. |
| 4 | THIF0 | Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested. |
| 3 | CFTIF0 | Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested. |
| 2 | TQIF0 | Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested. |
| 1 | TAIF0 | Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested. |
| 0 | TSIF0 | Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested. |

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCFDnCFDTMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the corresponding RSCFDnCFDTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCFDnCFDCmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCFDnCFDTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCFDnCFDTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCFDnCFDTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCFDnCFDTXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCFDnCFDCFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCFDnCFDCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCFDnCFDTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCFDnCFDTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCFDnCFDTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

17.4.4.7 RSCFDnCFDGFDCFG — Global FD Configuration Register

Access: RSCFDnCFDGFDCFG register can be read/written in 32-bit unit
 RSCFDnCFDGFDCFG L and RSCFDnCFDGFDCFG H registers can be read/written in 16-bit unit
 RSCFDnCFDGFDCFG LL, RSCFDnCFDGFDCFG LH, RSCFDnCFDGFDCFG HL, RSCFDnCFDGFDCFG HH registers can be read/written in 8-bit unit

Address: RSCFDnCFDGFDCFG: <RSCFDn_base> + 0474_H
 RSCFDnCFDGFDCFG L: <RSCFDn_base> + 0474_H, RSCFDnCFDGFDCFG H: <RSCFDn_base> + 0476_H
 RSCFDnCFDGFDCFG LL: <RSCFDn_base> + 0474_H, RSCFDnCFDGFDCFG LH: <RSCFDn_base> + 0475_H,
 RSCFDnCFDGFDCFG HL: <RSCFDn_base> + 0476_H, RSCFDnCFDGFDCFG HH: <RSCFDn_base> + 0477_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----|-----|-----|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TSCCFG[1:0] | — | — | — | — | — | — | — | — | RPED |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.112 RSCFDnCFDGFDCFG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 10 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 9, 8 | TSCCFG[1:0] | Time-stamp capture setting bit b9 b8 0 0: Captured at a sample point in the SOF bit. 0 1: Captured when a valid frame has been transmitted/received. 1 0: Captured at a sample point of the res bit. ^{Note 1} 1 1: Setting prohibited. |
| 7 to 1 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 0 | RPED | Protocol exception event detection disabled bit 0: Protocol exception event detection is enabled. 1: Protocol exception event detection is disabled. |

Note 1. When a Classical CAN frame is transmitted/received, a time-stamp value will be captured at the sample point in the SOF bit.

TSCCFG bit

Select a point where a time-stamp value is captured. Modify this bit only in global reset mode.

RPED bit

Setting this bit to 1 disables the protocol exception event detection. When a protocol exception event is detected while this bit is set to 1, the event is regarded as a form error and an error frame will be output. Modify this bit only in global reset mode.

17.4.5 Details of Receive Rule-related Registers

17.4.5.1 RSCFDnCFDGAFLECTR — Receive Rule Entry Control Register

Access: RSCFDnCFDGAFLECTR register can be read/written in 32-bit units
RSCFDnCFDGAFLECTRL, RSCFDnCFDGAFLECTRH registers can be read/written in 16-bit units
RSCFDnCFDGAFLECTRLL, RSCFDnCFDGAFLECTRLH, RSCFDnCFDGAFLECTRHL,
RSCFDnCFDGAFLECTRHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLECTR: <RSCFDn_base> + 0098_H
RSCFDnCFDGAFLECTRL: <RSCFDn_base> + 0098_H, RSCFDnCFDGAFLECTRH: <RSCFDn_base> + 009A_H
RSCFDnCFDGAFLECTRLL: <RSCFDn_base> + 0098_H, RSCFDnCFDGAFLECTRLH: <RSCFDn_base> + 0099_H,
RSCFDnCFDGAFLECTRHL: <RSCFDn_base> + 009A_H, RSCFDnCFDGAFLECTRHH: <RSCFDn_base> + 009B_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|------------|----|----|----|------------|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | AFLDA E | — | — | — | AFLPN[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 17.113 RSCFDnCFDGAFLECTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 9 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 8 | AFLDAE | Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled. |
| 7 to 5 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 4 to 0 | AFLPN[4:0] | Receive Rule Table Page Number Configuration A page number can be selected in the range of page 0 (0000 _B) to page 11 (01011 _B). |

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000_B to 01011_B.

17.4.5.2 RSCFDnCFDGAFLCFG0 — Receive Rule Configuration Register 0

Access: RSCFDnCFDGAFLCFG0 register can be read/written in 32-bit units
 RSCFDnCFDGAFLCFG0L, RSCFDnCFDGAFLCFG0H registers can be read/written in 16-bit units
 RSCFDnCFDGAFLCFG0LL, RSCFDnCFDGAFLCFG0LH, RSCFDnCFDGAFLCFG0HL,
 RSCFDnCFDGAFLCFG0HH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLCFG0: <RSCFDn_base> + 009C_H
 RSCFDnCFDGAFLCFG0L: <RSCFDn_base> + 009C_H, RSCFDnCFDGAFLCFG0H: <RSCFDn_base> +
 009E_H
 RSCFDnCFDGAFLCFG0LL: <RSCFDn_base> + 009C_H, RSCFDnCFDGAFLCFG0LH: <RSCFDn_base> +
 009D_H,
 RSCFDnCFDGAFLCFG0HL: <RSCFDn_base> + 009E_H, RSCFDnCFDGAFLCFG0HH: <RSCFDn_base> +
 009F_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----------|-----|-----|-----|-----|-----|-----|-----|
| | RNC0[7:0] | | | | | | | | RNC1[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RNC2[7:0] | | | | | | | | — | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

Table 17.114 RSCFDnCFDGAFLCFG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 24 | RNC0[7:0] | Number of Rules for Channel 0 Set the number of receive rules used for channel 0. |
| 23 to 16 | RNC1[7:0] | Number of Rules for Channel 1 Set the number of receive rules used for channel 1. |
| 15 to 8 | RNC2[7:0] | Number of Rules for Channel 2 Set the number of receive rules used for channel 2. |
| 7 to 0 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |

Modify the RSCFDnCFDGAFLCFG0 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total number of rules allocated to each channel does not exceed the number of rules that can be registered for the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 2 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

17.4.5.3 RSCFDnCFDGAFLIDj — Receive Rule ID Register j (j = 0 to 15)

Access: RSCFDnCFDGAFLIDj register can be read/written in 32-bit units
 RSCFDnCFDGAFLIDjL, RSCFDnCFDGAFLIDjH registers can be read/written in 16-bit units
 RSCFDnCFDGAFLIDjLL, RSCFDnCFDGAFLIDjLH, RSCFDnCFDGAFLIDjHL, RSCFDnCFDGAFLIDjHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLIDj: $\langle \text{RSCFDn_base} \rangle + 1000_{\text{H}} + (10_{\text{H}} \times j)$
 RSCFDnCFDGAFLIDjL: $\langle \text{RSCFDn_base} \rangle + 1000_{\text{H}} + (10_{\text{H}} \times j)$,
 RSCFDnCFDGAFLIDjH: $\langle \text{RSCFDn_base} \rangle + 1002_{\text{H}} + (10_{\text{H}} \times j)$
 RSCFDnCFDGAFLIDjLL: $\langle \text{RSCFDn_base} \rangle + 1000_{\text{H}} + (10_{\text{H}} \times j)$,
 RSCFDnCFDGAFLIDjLH: $\langle \text{RSCFDn_base} \rangle + 1001_{\text{H}} + (10_{\text{H}} \times j)$,
 RSCFDnCFDGAFLIDjHL: $\langle \text{RSCFDn_base} \rangle + 1002_{\text{H}} + (10_{\text{H}} \times j)$,
 RSCFDnCFDGAFLIDjHH: $\langle \text{RSCFDn_base} \rangle + 1003_{\text{H}} + (10_{\text{H}} \times j)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|-------------|------------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | GAFLID E | GAFLR TR | GAFL LB | GAFLID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.115 RSCFDnCFDGAFLIDj Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 | GAFLIDE | IDE Select 0: Standard ID 1: Extended ID |
| 30 | GAFLRTR | RTR Select 0: Data frame 1: Remote frame |
| 29 | GAFLLB | Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received |
| 28 to 0 | GAFLID[28:0] | ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0. |

Modify the RSCFDnCFDGAFLIDj register when the AFLDAE bit in the RSCFDnCFDGAFLLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

17.4.5.4 RSCFDnCFDGAFLMj — Receive Rule Mask Register j (j = 0 to 15)

Access: RSCFDnCFDGAFLMj register can be read/written in 32-bit units
 RSCFDnCFDGAFLMjL, RSCFDnCFDGAFLMjH registers can be read/written in 16-bit units
 RSCFDnCFDGAFLMjLL, RSCFDnCFDGAFLMjLH, RSCFDnCFDGAFLMjHL, RSCFDnCFDGAFLMjHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLMj: <RSCFDn_base> + 1004_H + (10_H × j)
 RSCFDnCFDGAFLMjL: <RSCFDn_base> + 1004_H + (10_H × j),
 RSCFDnCFDGAFLMjH: <RSCFDn_base> + 1006_H + (10_H × j)
 RSCFDnCFDGAFLMjLL: <RSCFDn_base> + 1004_H + (10_H × j),
 RSCFDnCFDGAFLMjLH: <RSCFDn_base> + 1005_H + (10_H × j),
 RSCFDnCFDGAFLMjHL: <RSCFDn_base> + 1006_H + (10_H × j),
 RSCFDnCFDGAFLMjHH: <RSCFDn_base> + 1007_H + (10_H × j)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|--------------|-----|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GAFLID EM | GAFLR TRM | — | GAFLIDM[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLIDM[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.116 RSCFDnCFDGAFLMj Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31 | GAFLIDEM | IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared. |
| 30 | GAFLRTRM | RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared. |
| 29 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 28 to 0 | GAFLIDM[28:0] | ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared. |

Modify the RSCFDnCFDGAFLMj register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCFDnCFDGAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set all the GAFLIDM[28:0] bits to 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

17.4.5.5 RSCFDnCFDGAFLP0_j — Receive Rule Pointer 0 Register j (j = 0 to 15)

Access: RSCFDnCFDGAFLP0j register can be read/written in 32-bit units
 RSCFDnCFDGAFLP0jL, RSCFDnCFDGAFLP0jH registers can be read/written in 16-bit units
 RSCFDnCFDGAFLP0jLL, RSCFDnCFDGAFLP0jLH, RSCFDnCFDGAFLP0jHL, RSCFDnCFDGAFLP0jHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLP0j: <RSCFDn_base> + 1008_H + (10_H × j)
 RSCFDnCFDGAFLP0jL: <RSCFDn_base> + 1008_H + (10_H × j),
 RSCFDnCFDGAFLP0jH: <RSCFDn_base> + 100A_H + (10_H × j)
 RSCFDnCFDGAFLP0jLL: <RSCFDn_base> + 1008_H + (10_H × j),
 RSCFDnCFDGAFLP0jLH: <RSCFDn_base> + 1009_H + (10_H × j),
 RSCFDnCFDGAFLP0jHL: <RSCFDn_base> + 100A_H + (10_H × j),
 RSCFDnCFDGAFLP0jHH: <RSCFDn_base> + 100B_H + (10_H × j)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|---------------|-----|-----|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GAFLDLC[3:0] | | | | GAFLPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLRMV | GAFLRMDP[6:0] | | | | | | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

Table 17.117 RSCFDnCFDGAFLP0_j Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---------------|---|-----|------------------------|---------------|-----|---------------------|--------------|---|---|---|---|------------------------|--|---|---|---|---|-------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|
| 31 to 28 | GAFLDLC[3:0] | Receive Rule DLC <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>b31</th> <th>b30</th> <th>b29</th> <th>b28</th> <th>Classical CAN Frame</th> <th>CAN FD Frame</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>DLC check is disabled.</td><td></td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 data byte</td><td></td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 data bytes</td><td></td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 data bytes</td><td></td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 data bytes</td><td></td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>8 data bytes</td><td>12 data bytes</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td></td><td>16 data bytes</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td></td><td>20 data bytes</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td></td><td>24 data bytes</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td></td><td>32 data bytes</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td></td><td>48 data bytes</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td>64 data bytes</td></tr> </tbody> </table> | b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | 0 | 0 | 0 | 0 | DLC check is disabled. | | 0 | 0 | 0 | 1 | 1 data byte | | 0 | 0 | 1 | 0 | 2 data bytes | | 0 | 0 | 1 | 1 | 3 data bytes | | 0 | 1 | 0 | 0 | 4 data bytes | | 0 | 1 | 0 | 1 | 5 data bytes | | 0 | 1 | 1 | 0 | 6 data bytes | | 0 | 1 | 1 | 1 | 7 data bytes | | 1 | 0 | 0 | 0 | 8 data bytes | | 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | 1 | 0 | 1 | 0 | | 16 data bytes | 1 | 0 | 1 | 1 | | 20 data bytes | 1 | 1 | 0 | 0 | | 24 data bytes | 1 | 1 | 0 | 1 | | 32 data bytes | 1 | 1 | 1 | 0 | | 48 data bytes | 1 | 1 | 1 | 1 | | 64 data bytes |
| b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | DLC check is disabled. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1 data byte | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 2 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 3 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 4 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 5 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 6 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 7 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 8 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | | 16 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | | 20 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | | 24 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 1 | | 32 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 0 | | 48 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | | 64 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 27 to 16 | GAFLPTR[11:0] | Receive Rule Label Set the 12-bit label information. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | GAFLRMV | Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 to 8 | GAFLRMDP[6:0] | Receive Buffer Number Select Set the receive buffer number to store receive messages. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 17.117 RSCFDnCFDGAFLP0_j Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 0 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |

Modify the RSCFDnCFDGAFLP0_j register when the AFLDAE bit in the RSCFDnCFDGAFLP0_j register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCFDnCFDRMNB register.

17.4.5.6 RSCFDnCFDGAFLP1_j — Receive Rule Pointer 1 Register j (j = 0 to 15)

Access: RSCFDnCFDGAFLP1_j register can be read/written in 32-bit units
 RSCFDnCFDGAFLP1_jL, RSCFDnCFDGAFLP1_jH registers can be read/written in 16-bit units
 RSCFDnCFDGAFLP1_jLL, RSCFDnCFDGAFLP1_jLH, RSCFDnCFDGAFLP1_jHL,
 RSCFDnCFDGAFLP1_jHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLP1_j: <RSCFDn_base> + 100C_H + (10_H × j)
 RSCFDnCFDGAFLP1_jL: <RSCFDn_base> + 100C_H + (10_H × j),
 RSCFDnCFDGAFLP1_jH: <RSCFDn_base> + 100E_H + (10_H × j)
 RSCFDnCFDGAFLP1_jLL: <RSCFDn_base> + 100C_H + (10_H × j),
 RSCFDnCFDGAFLP1_jLH: <RSCFDn_base> + 100D_H + (10_H × j),
 RSCFDnCFDGAFLP1_jHL: <RSCFDn_base> + 100E_H + (10_H × j),
 RSCFDnCFDGAFLP1_jHH: <RSCFDn_base> + 100F_H + (10_H × j)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | GAFLF DP16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GAFLFDP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.118 RSCFDnCFDGAFLP1_j Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 31 to 17 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 16 to 8 | GAFLFDP[16:8] | Transmit/Receive FIFO Buffer k Select (Bit position – 8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected. |
| 7 to 0 | GAFLFDP[7:0] | Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected. |

Modify the RSCFDnCFDGAFLP1_j register when the AFLDAE bit in the RSCFDnCFDGAFLFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP[16:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers can be selected. However, when the GAFLRMV bit in the RSCFDnCFDGAFLP0_j register is set to 1 (messages are stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCFDnCFDGFCCk register are set to 00_B (receive mode) or 10_B (gateway mode) can be selected.

17.4.6 Details of Receive Buffer-related Registers

17.4.6.1 RSCFDnCFDRMNB — Receive Buffer Number Register

Access: RSCFDnCFDRMNB register can be read/written in 32-bit units
 RSCFDnCFDRMNBL, RSCFDnCFDRMNBH registers can be read/written in 16-bit units
 RSCFDnCFDRMNBLL, RSCFDnCFDRMNBHL, RSCFDnCFDRMNBHL, RSCFDnCFDRMNBHH registers
 can be read/written in 8-bit units

Address: RSCFDnCFDRMNB: <RSCFDn_base> + 00A4_H
 RSCFDnCFDRMNBL: <RSCFDn_base> + 00A4_H, RSCFDnCFDRMNBH: <RSCFDn_base> + 00A6_H
 RSCFDnCFDRMNBLL: <RSCFDn_base> + 00A4_H, RSCFDnCFDRMNBHL: <RSCFDn_base> + 00A5_H,
 RSCFDnCFDRMNBHL: <RSCFDn_base> + 00A6_H, RSCFDnCFDRMNBHH: <RSCFDn_base> + 00A7_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|------------|-----|------------|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | RMPLS[1:0] | | NRXMB[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.119 RSCFDnCFDRMNB Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 8 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 9, 8 | RMPLS[1:0] | Receive Buffer Payload Storage Size Select b9 b8 0 0: 8 bytes 0 1: 12 bytes 1 0: 16 bytes 1 1: 20 bytes |
| 7 to 0 | NRXMB[7:0] | Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 48. |

Modify the RSCFDnCFDRMNB register only in global reset mode.

RMPLS[1:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive buffer.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CANFD module. The maximum value is $16 \times$ (number of channels).

Setting all of these bits to 0 makes receive buffers unavailable.

17.4.6.2 RSCFDnCFDRMNDy — Receive Buffer New Data Register y (y = 0, 1)

Access: RSCFDnCFDRMNDy register can be read/written in 32-bit units
 RSCFDnCFDRMNDyL, RSCFDnCFDRMNDyH registers can be read/written in 16-bit units
 RSCFDnCFDRMNDyLL, RSCFDnCFDRMNDyLH, RSCFDnCFDRMNDyHL, RSCFDnCFDRMNDyHH registers can be read/written in 8-bit units

Address: RSCFDnCFDRMNDy: <RSCFDn_base> + 00A8_H + (04_H × y)
 RSCFDnCFDRMNDyL: <RSCFDn_base> + 00A8_H + (04_H × y),
 RSCFDnCFDRMNDyH: <RSCFDn_base> + 00AA_H + (04_H × y)
 RSCFDnCFDRMNDyLL: <RSCFDn_base> + 00A8_H + (04_H × y),
 RSCFDnCFDRMNDyLH: <RSCFDn_base> + 00A9_H + (04_H × y),
 RSCFDnCFDRMNDyHL: <RSCFDn_base> + 00AA_H + (04_H × y),
 RSCFDnCFDRMNDyHH: <RSCFDn_base> + 00AB_H + (04_H × y)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.120 RSCFDnCFDRMNDy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | RMNSq | Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q. |
| 15 to 0 | RMNSq | Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q. |

Write 0 to the RSCFDnCFDRMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 47)

Each RMNSq flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. The message storing time depends on the storage payload size of the receive buffer. When the RMPLS[1:0] value in the RSCFDnCFDRMNB register is 00_B (8 bytes), the message storing time is 12 pclk clock cycles. When the RMPLS[1:0] value is 11_B (20 bytes), the message storing time is 18 pclk clock cycles. (2 pclk clock cycles per 4 bytes of storage payload size).

These flags are cleared to 0 in global reset mode.

17.4.6.3 RSCFDnCFDRMIDq — Receive Buffer ID Register q (q = 0 to 47)

Access: RSCFDnCFDRMIDq register can be read only in 32-bit units
 RSCFDnCFDRMIDqL, RSCFDnCFDRMIDqH registers can be read only in 16-bit units
 RSCFDnCFDRMIDqLL, RSCFDnCFDRMIDqLH, RSCFDnCFDRMIDqHL, RSCFDnCFDRMIDqHH registers can be read only in 8-bit units

Address: RSCFDnCFDRMIDq: <RSCFDn_base> + 2000_H + (20_H × q)
 RSCFDnCFDRMIDqL: <RSCFDn_base> + 2000_H + (20_H × q),
 RSCFDnCFDRMIDqH: <RSCFDn_base> + 2002_H + (20_H × q)
 RSCFDnCFDRMIDqLL: <RSCFDn_base> + 2000_H + (20_H × q),
 RSCFDnCFDRMIDqLH: <RSCFDn_base> + 2001_H + (20_H × q),
 RSCFDnCFDRMIDqHL: <RSCFDn_base> + 2002_H + (20_H × q),
 RSCFDnCFDRMIDqHH: <RSCFDn_base> + 2003_H + (20_H × q)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-------|----|-------------|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RMIDE | RMRTR | — | RMID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.121 RSCFDnCFDRMIDq Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 | RMIDE | Receive Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | RMRTR | Receive Buffer RTR/RRS <ul style="list-style-type: none"> When the received message is a classical CAN frame 0: Data frame 1: Remote frame When the received message is a CAN FD frame The RRS bit value of the received message can be read. |
| 29 | Reserved | These bits are read as the value after reset. |
| 28 to 0 | RMID[28:0] | Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0. |

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer. When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

17.4.6.4 RSCFDnCFDRMPTRq — Receive Buffer Pointer Register q (q = 0 to 47)

Access: RSCFDnCFDRMPTRq register can be read only in 32-bit units
 RSCFDnCFDRMPTRqL, RSCFDnCFDRMPTRqH registers can be read only in 16-bit units
 RSCFDnCFDRMPTRqLL, RSCFDnCFDRMPTRqLH, RSCFDnCFDRMPTRqHL, RSCFDnCFDRMPTRqHH registers can be read only in 8-bit units

Address: RSCFDnCFDRMPTRq: <RSCFDn_base> + 2004_H + (20_H × q)
 RSCFDnCFDRMPTRqL: <RSCFDn_base> + 2004_H + (20_H × q),
 RSCFDnCFDRMPTRqH: <RSCFDn_base> + 2006_H + (20_H × q)
 RSCFDnCFDRMPTRqLL: <RSCFDn_base> + 2004_H + (20_H × q),
 RSCFDnCFDRMPTRqLH: <RSCFDn_base> + 2005_H + (20_H × q),
 RSCFDnCFDRMPTRqHL: <RSCFDn_base> + 2006_H + (20_H × q),
 RSCFDnCFDRMPTRqHH: <RSCFDn_base> + 2007_H + (20_H × q)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|-------------|----|----|----|----|----|----|----|----|----|----|----|
| | RMDLC[3:0] | | | | RMPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.122 RSCFDnCFDRMPTRq Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|-------------|---|-----|---------------------|---------------|-----|---------------------|--------------|---|---|---|---|--------------|--|---|---|---|---|-------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|
| 31 to 28 | RMDLC[3:0] | Receive Buffer DLC Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>b31</th> <th>b30</th> <th>b29</th> <th>b28</th> <th>Classical CAN Frame</th> <th>CAN FD Frame</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1 data byte</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>2 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>3 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>4 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>5 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>6 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>7 data bytes</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>8 data bytes</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>8 data bytes</td> <td>12 data bytes</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td></td> <td>16 data bytes</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td></td> <td>20 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td></td> <td>24 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td></td> <td>32 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td></td> <td>48 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td></td> <td>64 data bytes</td> </tr> </tbody> </table> | b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | 0 | 0 | 0 | 0 | 0 data bytes | | 0 | 0 | 0 | 1 | 1 data byte | | 0 | 0 | 1 | 0 | 2 data bytes | | 0 | 0 | 1 | 1 | 3 data bytes | | 0 | 1 | 0 | 0 | 4 data bytes | | 0 | 1 | 0 | 1 | 5 data bytes | | 0 | 1 | 1 | 0 | 6 data bytes | | 0 | 1 | 1 | 1 | 7 data bytes | | 1 | 0 | 0 | 0 | 8 data bytes | | 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | 1 | 0 | 1 | 0 | | 16 data bytes | 1 | 0 | 1 | 1 | | 20 data bytes | 1 | 1 | 0 | 0 | | 24 data bytes | 1 | 1 | 0 | 1 | | 32 data bytes | 1 | 1 | 1 | 0 | | 48 data bytes | 1 | 1 | 1 | 1 | | 64 data bytes |
| b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1 data byte | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 2 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 3 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 4 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 5 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 6 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 7 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 8 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | | 16 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | | 20 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | | 24 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 1 | | 32 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 0 | | 48 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | | 64 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 27 to 16 | RMPTR[11:0] | Receive Buffer Label Data Label information of the received message. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 to 0 | RMTS[15:0] | Receive Buffer Timestamp Data Timestamp value of the received message. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer. The number of bytes of the payload to be stored in the receive buffer is determined by the RMPLS[1:0] bits in the RSCFDnCFDRMNB register.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

17.4.6.5 RSCFDnCFDRMFDSTSq — Receive Buffer CAN FD Status Register q (q = 0 to 47)

Access: RSCFDnCFDRMFDSTSq register can be read only in 32-bit units
 RSCFDnCFDRMFDSTSqL, RSCFDnCFDRMFDSTSqH registers can be read only in 16-bit units
 RSCFDnCFDRMFDSTSqLL, RSCFDnCFDRMFDSTSqLH, RSCFDnCFDRMFDSTSqHL,
 RSCFDnCFDRMFDSTSqHH registers can be read only in 8-bit units

Address: RSCFDnCFDRMFDSTSq: <RSCFDn_base> + 2008_H + (20_H × q)
 RSCFDnCFDRMFDSTSqL: <RSCFDn_base> + 2008_H + (20_H × q),
 RSCFDnCFDRMFDSTSqH: <RSCFDn_base> + 200A_H + (20_H × q)
 RSCFDnCFDRMFDSTSqLL: <RSCFDn_base> + 2008_H + (20_H × q),
 RSCFDnCFDRMFDSTSqLH: <RSCFDn_base> + 2009_H + (20_H × q),
 RSCFDnCFDRMFDSTSqHL: <RSCFDn_base> + 200A_H + (20_H × q),
 RSCFDnCFDRMFDSTSqHH: <RSCFDn_base> + 200B_H + (20_H × q)

Initial value: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RMFDF | RMBRS | RMESI |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.123 RSCFDnCFDRMFDSTSq Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | These bits are read as the value after reset. |
| 2 | RMFDF | FD Format 0: Classical CAN frame 1: CAN FD frame |
| 1 | RMBRS | BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes. |
| 0 | RMESI | ESI 0: Error active node 1: Error passive node |

RMFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive buffer.

RMBRS Bit

When the RMFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the receive buffer. When the RMFDF bit is set to 0, this bit is always read as 0.

RMESI Bit

When the RMFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the receive buffer. When the RMFDF bit is set to 0, this bit is always read as 0.

17.4.6.6 RSCFDnCFDRMDFb_q — Receive Buffer Data Field b Register q (b = 0 to 4, q = 0 to 47)

Access: RSCFDnCFDRMDFb_q register can be read only in 32-bit units
RSCFDnCFDRMDFb_qL, RSCFDnCFDRMDFb_qH registers can be read only in 16-bit units
RSCFDnCFDRMDFb_qLL, RSCFDnCFDRMDFb_qLH, RSCFDnCFDRMDFb_qHL,
RSCFDnCFDRMDFb_qHH registers can be read only in 8-bit units

Address: RSCFDnCFDRMDFb_q: $\langle \text{RSCFDn_base} \rangle + 200C_H + (04_H \times b) + (20_H \times q)$
RSCFDnCFDRMDFb_qL: $\langle \text{RSCFDn_base} \rangle + 200C_H + (04_H \times b) + (20_H \times q)$,
RSCFDnCFDRMDFb_qH: $\langle \text{RSCFDn_base} \rangle + 200E_H + (04_H \times b) + (20_H \times q)$
RSCFDnCFDRMDFb_qLL: $\langle \text{RSCFDn_base} \rangle + 200C_H + (04_H \times b) + (20_H \times q)$,
RSCFDnCFDRMDFb_qLH: $\langle \text{RSCFDn_base} \rangle + 200D_H + (04_H \times b) + (20_H \times q)$,
RSCFDnCFDRMDFb_qHL: $\langle \text{RSCFDn_base} \rangle + 200E_H + (04_H \times b) + (20_H \times q)$,
RSCFDnCFDRMDFb_qHH: $\langle \text{RSCFDn_base} \rangle + 200F_H + (04_H \times b) + (20_H \times q)$

Initial value: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------------|----|----|----|----|----|----|----|---------------------|----|----|----|----|----|----|----|
| | RMDB4 × b + 3 [7:0] | | | | | | | | RMDB4 × b + 2 [7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RMDB4 × b + 1 [7:0] | | | | | | | | RMDB4 × b + 0 [7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.124 RSCFDnCFDRMDFb_q Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|--|
| 31 to 24 | RMDB4 × b + 3 [7:0] | Receive Buffer Data Byte 4 × b + 3 |
| | | Receive Buffer Data Byte 4 × b + 2 |
| 23 to 16 | RMDB4 × b + 2 [7:0] | Receive Buffer Data Byte 4 × b + 1 |
| | | Receive Buffer Data Byte 4 × b + 0 |
| 15 to 8 | RMDB4 × b + 1 [7:0] | Data for a message stored in the receive buffer can be read. |
| 7 to 0 | RMDB4 × b + 0 [7:0] | |

When the RMDLC[3:0] value in the RSCFDnCFDRMPTRq register is smaller than the payload storage size of the receive buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the receive buffer by the RMPLS[1:0] bits in the RSCFDnCFDRMNB register. Do not read or write the RSCFDnCFDRMDFb_q register corresponding to an area that exceeds the specified size.

17.4.7 Details of Receive FIFO Buffer-related Registers

17.4.7.1 RSCFDnCFDRFCCx — Receive FIFO Buffer Configuration and Control Register x (x = 0 to 7)

Access: RSCFDnCFDRFCCx register can be read/written in 32-bit units
 RSCFDnCFDRFCCxL, RSCFDnCFDRFCCxH registers can be read/written in 16-bit units
 RSCFDnCFDRFCCxLL, RSCFDnCFDRFCCxLH, RSCFDnCFDRFCCxHL, RSCFDnCFDRFCCxHH registers can be read/written in 8-bit units

Address: RSCFDnCFDRFCCx: <RSCFDn_base> + 00B8_H + (04_H × x)
 RSCFDnCFDRFCCxL: <RSCFDn_base> + 00B8_H + (04_H × x),
 RSCFDnCFDRFCCxH: <RSCFDn_base> + 00BA_H + (04_H × x)
 RSCFDnCFDRFCCxLL: <RSCFDn_base> + 00B8_H + (04_H × x),
 RSCFDnCFDRFCCxLH: <RSCFDn_base> + 00B9_H + (04_H × x),
 RSCFDnCFDRFCCxHL: <RSCFDn_base> + 00BA_H + (04_H × x),
 RSCFDnCFDRFCCxHH: <RSCFDn_base> + 00BB_H + (04_H × x)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|-----|-----|------|----|-----------|-----|-----|----|------------|-----|-----|----|----|------|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFIGCV[2:0] | | | RFIM | — | RFDC[2:0] | | | — | RFPLS[2:0] | | | — | — | RFIE | RFE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R | R/W | R/W |

Table 17.125 RSCFDnCFDRFCCx Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 16 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 15 to 13 | RFIGCV[2:0] | Receive FIFO Interrupt Request Timing Select $b_{15} \ b_{14} \ b_{13}$ 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full. |
| 12 | RFIM | Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received. |
| 11 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 10 to 8 | RFDC[2:0] | Receive FIFO Buffer Depth Configuration $b_{10} \ b_9 \ b_8$ 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages |

Table 17.125 RSCFDnCFDRFCCx Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 7 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 6 to 4 | RFPLS[2:0] | Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes |
| 3, 2 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 1 | RFIE | Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled. |
| 0 | RFE | Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used. |

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive FIFO buffer. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit is set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 enables the use of FIFO buffers. Clearing this bit to 0 sets the RFEMP flag in the RSCFDnCFDRFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode. Set this bit to 1 with another instruction after the settings to all bits in the RSCFDnCFDRFCCx register have been done. This bit is cleared to 0 in global reset mode.

17.4.7.2 RSCFDnCFDRFSTSx — Receive FIFO Buffer Status Register x (x = 0 to 7)

Access: RSCFDnCFDRFSTSx register can be read/written in 32-bit units
 RSCFDnCFDRFSTSxL, RSCFDnCFDRFSTSxH registers can be read/written in 16-bit units
 RSCFDnCFDRFSTSxLL, RSCFDnCFDRFSTSxLH, RSCFDnCFDRFSTSxHL, RSCFDnCFDRFSTSxHH registers can be read/written in 8-bit units

Address: RSCFDnCFDRFSTSx: <RSCFDn_base> + 00D8_H + (04_H × x)
 RSCFDnCFDRFSTSxL: <RSCFDn_base> + 00D8_H + (04_H × x),
 RSCFDnCFDRFSTSxH: <RSCFDn_base> + 00DA_H + (04_H × x)
 RSCFDnCFDRFSTSxLL: <RSCFDn_base> + 00D8_H + (04_H × x),
 RSCFDnCFDRFSTSxLH: <RSCFDn_base> + 00D9_H + (04_H × x),
 RSCFDnCFDRFSTSxHL: <RSCFDn_base> + 00DA_H + (04_H × x),
 RSCFDnCFDRFSTSxHH: <RSCFDn_base> + 00DB_H + (04_H × x)

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|------|-------|-------|-------|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFMC[7:0] | | | | | | | — | — | — | — | RFIF | RFMLT | RFFLL | RFEMP | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.126 RSCFDnCFDRFSTSx Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 15 to 8 | RFMC[7:0] | Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed. |
| 7 to 4 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 3 | RFIF | Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present. |
| 2 | RFMLT | Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost. |
| 1 | RFFLL | Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full. |
| 0 | RFEMP | Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty). |

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RSCFDnCFDRFCCx register is set to 0.

This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCFDnCFDRFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flags to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCFDnCFDRFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCFDnCFDRFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCFDnCFDRFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when a received message is in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

17.4.7.3 RSCFDnCFDRFPCTR_x — Receive FIFO Buffer Pointer Control Register _x (x = 0 to 7)

Access: RSCFDnCFDRFPCTR_x register can only be written in 32-bit units
RSCFDnCFDRFPCTR_{xL}, RSCFDnCFDRFPCTR_{xH} registers can only be written in 16-bit units
RSCFDnCFDRFPCTR_{xLL}, RSCFDnCFDRFPCTR_{xLH}, RSCFDnCFDRFPCTR_{xHL},
RSCFDnCFDRFPCTR_{xHH} registers can only be written in 8-bit units

Address: RSCFDnCFDRFPCTR_x: <RSCFDn_base> + 00F8_H + (04_H × x)
RSCFDnCFDRFPCTR_{xL}: <RSCFDn_base> + 00F8_H + (04_H × x),
RSCFDnCFDRFPCTR_{xH}: <RSCFDn_base> + 00FA_H + (04_H × x)
RSCFDnCFDRFPCTR_{xLL}: <RSCFDn_base> + 00F8_H + (04_H × x),
RSCFDnCFDRFPCTR_{xLH}: <RSCFDn_base> + 00F9_H + (04_H × x),
RSCFDnCFDRFPCTR_{xHL}: <RSCFDn_base> + 00FA_H + (04_H × x),
RSCFDnCFDRFPCTR_{xHH}: <RSCFDn_base> + 00FB_H + (04_H × x)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | RFPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.127 RSCFDnCFDRFPCTR_x Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 8 | Reserved | The write value should be the value after reset. |
| 7 to 0 | RFPC[7:0] | Receive FIFO Pointer Control When these bits are set to FF _H , the read pointer moves to the next unread message in the receive FIFO buffer. |

When the RFDMAEx value in the RSCFDnCFDCDTCT register is 1 (DMA transfer request enabled), do not write a value to this register.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCFDnCFDRFSTSt_x register is decremented by 1. Read the RSCFDnCFDRFID_x, RSCFDnCFDRFPTR_x, RSCFDnCFDRFFDSTSt_x, and RSCFDnCFDRFDFd_x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RSCFDnCFDRFCC_x register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCFDnCFDRFSTSt_x register is 0 (the receive FIFO buffer contains unread messages).

17.4.7.4 RSCFDnCFDRFIDx — Receive FIFO Buffer Access ID Register x (x = 0 to 7)

Access: RSCFDnCFDRFIDx register can be read only in 32-bit units
 RSCFDnCFDRFIDxL, RSCFDnCFDRFIDxH registers can be read only in 16-bit units
 RSCFDnCFDRFIDxLL, RSCFDnCFDRFIDxLH, RSCFDnCFDRFIDxHL, RSCFDnCFDRFIDxHH registers can be read only in 8-bit units

Address: RSCFDnCFDRFIDx: <RSCFDn_base> + 3000_H + (80_H × x)
 RSCFDnCFDRFIDxL: <RSCFDn_base> + 3000_H + (80_H × x),
 RSCFDnCFDRFIDxH: <RSCFDn_base> + 3002_H + (80_H × x)
 RSCFDnCFDRFIDxLL: <RSCFDn_base> + 3000_H + (80_H × x),
 RSCFDnCFDRFIDxLH: <RSCFDn_base> + 3001_H + (80_H × x),
 RSCFDnCFDRFIDxHL: <RSCFDn_base> + 3002_H + (80_H × x),
 RSCFDnCFDRFIDxHH: <RSCFDn_base> + 3003_H + (80_H × x)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|----|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RFIDE | | RFRTR | RFID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.128 RSCFDnCFDRFIDx Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 | RFIDE | Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | RFRTR | Receive FIFO Buffer RTR/RRS <ul style="list-style-type: none"> When the received message is a classical CAN frame 0: Data frame 1: Remote frame When the received message is a CAN FD frame The RRS bit value of the received message can be read. |
| 29 | Reserved | These bits are read as the value after reset. |
| 28 to 0 | RFID[28:0] | Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0. |

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer. When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

17.4.7.5 RSCFDnCFDRFPTRx — Receive FIFO Buffer Access Pointer Register x (x = 0 to 7)

Access: RSCFDnCFDRFPTRx register can be read only in 32-bit units
 RSCFDnCFDRFPTRxL, RSCFDnCFDRFPTRxH registers can be read only in 16-bit units
 RSCFDnCFDRFPTRxLL, RSCFDnCFDRFPTRxLH, RSCFDnCFDRFPTRxHL, RSCFDnCFDRFPTRxHH registers can be read only in 8-bit units

Address: RSCFDnCFDRFPTRx: <RSCFDn_base> + 3004_H + (80_H × x)
 RSCFDnCFDRFPTRxL: <RSCFDn_base> + 3004_H + (80_H × x),
 RSCFDnCFDRFPTRxH: <RSCFDn_base> + 3006_H + (80_H × x)
 RSCFDnCFDRFPTRxLL: <RSCFDn_base> + 3004_H + (80_H × x),
 RSCFDnCFDRFPTRxLH: <RSCFDn_base> + 3005_H + (80_H × x),
 RSCFDnCFDRFPTRxHL: <RSCFDn_base> + 3006_H + (80_H × x),
 RSCFDnCFDRFPTRxHH: <RSCFDn_base> + 3007_H + (80_H × x)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|-------------|----|----|----|----|----|----|----|----|----|----|----|
| | RFDLC[3:0] | | | | RFPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.129 RSCFDnCFDRFPTRx Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|-------------|---|-----|---------------------|---------------|-----|---------------------|--------------|---|---|---|---|--------------|--|---|---|---|---|-------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|
| 31 to 28 | RFDLC[3:0] | Receive FIFO Buffer DLC Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>b31</th> <th>b30</th> <th>b29</th> <th>b28</th> <th>Classical CAN Frame</th> <th>CAN FD Frame</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0 data bytes</td><td></td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 data byte</td><td></td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 data bytes</td><td></td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 data bytes</td><td></td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 data bytes</td><td></td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 data bytes</td><td></td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>8 data bytes</td><td>12 data bytes</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td></td><td>16 data bytes</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td></td><td>20 data bytes</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td></td><td>24 data bytes</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td></td><td>32 data bytes</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td></td><td>48 data bytes</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td>64 data bytes</td></tr> </tbody> </table> | b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | 0 | 0 | 0 | 0 | 0 data bytes | | 0 | 0 | 0 | 1 | 1 data byte | | 0 | 0 | 1 | 0 | 2 data bytes | | 0 | 0 | 1 | 1 | 3 data bytes | | 0 | 1 | 0 | 0 | 4 data bytes | | 0 | 1 | 0 | 1 | 5 data bytes | | 0 | 1 | 1 | 0 | 6 data bytes | | 0 | 1 | 1 | 1 | 7 data bytes | | 1 | 0 | 0 | 0 | 8 data bytes | | 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | 1 | 0 | 1 | 0 | | 16 data bytes | 1 | 0 | 1 | 1 | | 20 data bytes | 1 | 1 | 0 | 0 | | 24 data bytes | 1 | 1 | 0 | 1 | | 32 data bytes | 1 | 1 | 1 | 0 | | 48 data bytes | 1 | 1 | 1 | 1 | | 64 data bytes |
| b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1 data byte | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 2 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 3 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 4 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 5 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 6 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 7 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 8 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | | 16 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | | 20 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | | 24 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 1 | | 32 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 0 | | 48 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | | 64 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 27 to 16 | RFPTR[11:0] | Receive FIFO Buffer Label Data Label information of the received message can be read. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 to 0 | RFTS[15:0] | Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

17.4.7.6 RSCFDnCFDRFFDSTSx — Receive FIFO CAN FD Status Register x (x = 0 to 7)

Access: RSCFDnCFDRFFDSTSx register can be read only in 32-bit units
 RSCFDnCFDRFFDSTSxL, RSCFDnCFDRFFDSTSxH registers can be read only in 16-bit units
 RSCFDnCFDRFFDSTSxLL, RSCFDnCFDRFFDSTSxLH, RSCFDnCFDRFFDSTSxHL,
 RSCFDnCFDRFFDSTSxHH registers can be read only in 8-bit units

Address: RSCFDnCFDRFFDSTSx: <RSCFDn_base> + 3008_H + (80_H × x)
 RSCFDnCFDRFFDSTSxL: <RSCFDn_base> + 3008_H + (80_H × x),
 RSCFDnCFDRFFDSTSxH: <RSCFDn_base> + 300A_H + (80_H × x)
 RSCFDnCFDRFFDSTSxLL: <RSCFDn_base> + 3008_H + (80_H × x),
 RSCFDnCFDRFFDSTSxLH: <RSCFDn_base> + 3009_H + (80_H × x),
 RSCFDnCFDRFFDSTSxHL: <RSCFDn_base> + 300A_H + (80_H × x),
 RSCFDnCFDRFFDSTSxHH: <RSCFDn_base> + 300B_H + (80_H × x)

Initial value: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RFFDF | RFBRS | RFESI |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.130 RSCFDnCFDRFFDSTSx Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | These bits are read as the value after reset. |
| 2 | RFFDF | FD Format 0: Classical CAN frame 1: CAN FD frame |
| 1 | RFBRS | BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes. |
| 0 | RFESI | ESI 0: Error active node 1: Error passive node |

RFFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive FIFO buffer.

RFBRS Bit

When the RFFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is set to 0, this bit is always read as 0.

RFESI Bit

When the RFFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is set to 0, this bit is always read as 0.

17.4.7.7 RSCFDnCFDRFDFd_x — Receive FIFO Buffer Access Data Field d Register x (d = 0 to 15, x = 0 to 7)

Access: RSCFDnCFDRFDFd_x register can be read only in 32-bit units
 RSCFDnCFDRFDFdx_LL, RSCFDnCFDRFDFdx_H registers can be read only in 16-bit units
 RSCFDnCFDRFDFdx_LL, RSCFDnCFDRFDFdx_LH, RSCFDnCFDRFDFdx_HL, RSCFDnCFDRFDFdx_HH registers can be read only in 8-bit units

Address: RSCFDnCFDRFDFd_x: <RSCFDn_base> + 300C_H + (04_H × d) + (80_H × x)
 RSCFDnCFDRFDFd_xL: <RSCFDn_base> + 300C_H + (04_H × d) + (80_H × x),
 RSCFDnCFDRFDFd_xH: <RSCFDn_base> + 300E_H + (04_H × d) + (80_H × x)
 RSCFDnCFDRFDFd_xLL: <RSCFDn_base> + 300C_H + (04_H × d) + (80_H × x),
 RSCFDnCFDRFDFd_xLH: <RSCFDn_base> + 300D_H + (04_H × d) + (80_H × x),
 RSCFDnCFDRFDFd_xHL: <RSCFDn_base> + 300E_H + (04_H × d) + (80_H × x),
 RSCFDnCFDRFDFd_xHH: <RSCFDn_base> + 300F_H + (04_H × d) + (80_H × x)

Initial value: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------------|----|----|----|----|----|----|----|---------------------|----|----|----|----|----|----|----|
| | RFDB4 × d + 3 [7:0] | | | | | | | | RFDB4 × d + 2 [7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFDB4 × d + 1 [7:0] | | | | | | | | RFDB4 × d + 0 [7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.131 RSCFDnCFDRFDFd_x Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|--|
| 31 to 24 | RFDB4 × d + 3 [7:0] | Receive Buffer Data Byte 4 × d + 3 |
| | | Receive Buffer Data Byte 4 × d + 2 |
| 23 to 16 | RFDB4 × d + 2 [7:0] | Receive Buffer Data Byte 4 × d + 1 |
| | | Receive Buffer Data Byte 4 × d + 0 |
| 15 to 8 | RFDB4 × d + 1 [7:0] | Data for a message stored in the receive buffer can be read. |
| 7 to 0 | RFDB4 × d + 0 [7:0] | |

When the RFDLC[3:0] value in the RSCFDnCFDRFPTRx register is smaller than the payload storage size of the receive FIFO buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the receive FIFO buffer by the RFPLS[2:0] bits in the RSCFDnCFDRFCCx register. Do not read or write the RSCFDnCFDRFDFd_x register corresponding to an area that exceeds the specified size.

17.4.8 Transmit/Receive FIFO Buffer Related Registers

17.4.8.1 RSCFDnCFDCFCCK — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 8)

Access: RSCFDnCFDCFCCK register can be read/written in 32-bit units
 RSCFDnCFDCFCCKL, RSCFDnCFDCFCCKH registers can be read/written in 16-bit units
 RSCFDnCFDCFCCKLL, RSCFDnCFDCFCCKLH, RSCFDnCFDCFCCKHL, RSCFDnCFDCFCCKHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFCCK: <RSCFDn_base> + 0118_H + (04_H × k)
 RSCFDnCFDCFCCKL: <RSCFDn_base> + 0118_H + (04_H × k),
 RSCFDnCFDCFCCKH: <RSCFDn_base> + 011A_H + (04_H × k)
 RSCFDnCFDCFCCKLL: <RSCFDn_base> + 0118_H + (04_H × k),
 RSCFDnCFDCFCCKLH: <RSCFDn_base> + 0119_H + (04_H × k),
 RSCFDnCFDCFCCKHL: <RSCFDn_base> + 011A_H + (04_H × k),
 RSCFDnCFDCFCCKHH: <RSCFDn_base> + 011B_H + (04_H × k)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|-----|-----|------|-----|-----------|-----|-----|------------|-----|-----|-------|--------|----------|-----|-----|
| | CFITT[7:0] | | | | | | | | CFTML[3:0] | | | CFITR | CFITSS | CFM[1:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFIGCV[2:0] | | | CFIM | — | CFDC[2:0] | | — | CFPLS[2:0] | | | — | CFTXIE | CFRXIE | CFE | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Table 17.132 RSCFDnCFDCFCCK Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 24 | CFITT[7:0] | Set a message transmission interval. Set Value: 00 _H to FF _H |
| 23 to 20 | CFTML[3:0] | Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer. |
| 19 | CFITR | Transmit/Receive FIFO Interval Timer Resolution 0: Clock obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits 1: Clock obtained by dividing pclk/2 by the “value of ITRCP[15:0] bits × 10” |
| 18 | CFITSS | Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the nominal bit time clock for the channel to which the FIFO is linked. |
| 17, 16 | CFM[1:0] | Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited |
| 15 to 13 | CFIGCV[2:0] | Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full. |

Table 17.132 RSCFDnCFDCFCCK Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 12 | CFIM | Transmit/Receive FIFO Interrupt Source Select 0: <ul style="list-style-type: none"> Receive mode/gateway mode When the number of received messages meets the condition set by the CFICV[2:0] bits, a FIFO receive interrupt request is generated. Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: <ul style="list-style-type: none"> Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted. |
| 11 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 10 to 8 | CFDC[2:0] | Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages |
| 7 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 6 to 4 | CFPLS[2:0] | Transmit/Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes |
| 3 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 2 | CFTXIE | Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled. |
| 1 | CFRXIE | Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled. |
| 0 | CFE | Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used. |

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as $m = k/3$ (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be $((16 \times m) + CFTML[3:0])$ (see **Table 17.92**).

See **Table 17.90** and **Table 17.91**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p .

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the same channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the $pclk/2$ clock divided by the value of the ITRCP[15:0] bits in the RSCFDnCFDGCFCFG register.

When this bit is 1, the interval timer clock source is the $pclk/2$ clock divided by (the value of the ITRCP[15:0] bits in the RSCFDnCFDGCFCFG register $\times 10$).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the nominal bit time clock of the channel to which the FIFO is linked is the count source of the interval timer. Use this count source only for the channel does not handle the CAN FD frames.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages required to generate a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B , 011_B , 101_B , or 111_B .

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCFDnCFDCFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCFDnCFDCFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 enables the transmit/receive FIFO buffers.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCFDnCFDCFCCK register have been set, set this bit to 1 by using another instruction.

17.4.8.2 RSCFDnCFDCFSTSk — Transmit/receive FIFO Buffer Status Register k (k = 0 to 8)

Access: RSCFDnCFDCFSTSk register can be read/written in 32-bit units
 RSCFDnCFDCFSTSkL, RSCFDnCFDCFSTSkH registers can be read/written in 16-bit units
 RSCFDnCFDCFSTSkLL, RSCFDnCFDCFSTSkLH, RSCFDnCFDCFSTSkHL, RSCFDnCFDCFSTSkHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFSTSk: <RSCFDn_base> + 0178_H + (04_H × k)
 RSCFDnCFDCFSTSkL: <RSCFDn_base> + 0178_H + (04_H × k),
 RSCFDnCFDCFSTSkH: <RSCFDn_base> + 017A_H + (04_H × k)
 RSCFDnCFDCFSTSkLL: <RSCFDn_base> + 0178_H + (04_H × k),
 RSCFDnCFDCFSTSkLH: <RSCFDn_base> + 0179_H + (04_H × k),
 RSCFDnCFDCFSTSkHL: <RSCFDn_base> + 017A_H + (04_H × k),
 RSCFDnCFDCFSTSkHH: <RSCFDn_base> + 017B_H + (04_H × k)

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|--------|--------|-------|-------|-------|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFMC[7:0] | | | | | | | — | — | — | CFTXIF | CFRXIF | CFMLT | CFLL | CFEMP | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.133 RSCFDnCFDCFSTSk Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 15 to 8 | CFMC[7:0] | Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer. |
| 7 to 5 | Reserved | These bits are read as the value after reset. When writing, write the value after reset. |
| 4 | CFTXIF | Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present. |
| 3 | CFRXIF | Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present. |
| 2 | CFMLT | Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost. |
| 1 | CFLL | Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full. |
| 0 | CFEMP | Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty). |

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values according to the setting of the CFM[1:0] bits in the RSCFDnCFDCFCCK register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode
- When the CFE bit in the RSCFDnCFDCFCCK register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the interrupt source selected by the CFIM bit in the RSCFDnCFDCFCCK register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RSCFDnCFDCFCCK register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFLL Flag

The CFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCFDnCFDCFCCK register.

The CFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCFDnCFDCFCCK register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RSCFDnCFDCFPCTRk register after data was written to the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDk registers.

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

17.4.8.3 RSCFDnCFDCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register k (k = 0 to 8)

Access: RSCFDnCFDCFPCTRk register can only be written in 32-bit units
RSCFDnCFDCFPCTRkL, RSCFDnCFDCFPCTRkH registers can only be written in 16-bit units
RSCFDnCFDCFPCTRkLL, RSCFDnCFDCFPCTRkLH, RSCFDnCFDCFPCTRkHL,
RSCFDnCFDCFPCTRkHH registers can only be written in 8-bit units

Address: RSCFDnCFDCFPCTRk: $\langle \text{RSCFDn_base} \rangle + 01D8_H + (04_H \times k)$
RSCFDnCFDCFPCTRkL: $\langle \text{RSCFDn_base} \rangle + 01D8_H + (04_H \times k)$,
RSCFDnCFDCFPCTRkH: $\langle \text{RSCFDn_base} \rangle + 01DA_H + (04_H \times k)$
RSCFDnCFDCFPCTRkLL: $\langle \text{RSCFDn_base} \rangle + 01D8_H + (04_H \times k)$,
RSCFDnCFDCFPCTRkLH: $\langle \text{RSCFDn_base} \rangle + 01D9_H + (04_H \times k)$,
RSCFDnCFDCFPCTRkHL: $\langle \text{RSCFDn_base} \rangle + 01DA_H + (04_H \times k)$,
RSCFDnCFDCFPCTRkHH: $\langle \text{RSCFDn_base} \rangle + 01DB_H + (04_H \times k)$

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | CFPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.134 RSCFDnCFDCFPCTRk Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 8 | Reserved | The write value should be the value after reset. |
| 7 to 0 | CFPC[7:0] | Transmit/Receive FIFO Pointer Control <ul style="list-style-type: none"> Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. Gateway mode: Setting prohibited |

When the corresponding transmit/receive FIFO buffer is the first transmit/receive FIFO buffer (k = 3 × m) allocated to channel m and when the CFDMAEm bit in the RSCFDnCFDCDTCT register is 1 (DMA transfer request enabled), do not write a value to this register.

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCFDnCFDCFCCK register is 00_B):
Writing FF_H to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCFDnCFDCFSTSk register is decremented by 1. Read the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDf_k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.
When writing FF_H to these bits, make sure that the CFE bit in the RSCFDnCFDCFCCK register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCFDnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).
- Transmit mode (CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B):
Writing FF_H to the CFPC[7:0] bits stores the data written to the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDf_k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1. Write transmit messages to the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDf_k registers before writing FF_H to the CFPC[7:0] bits.
When writing FF_H to these bits, make sure that the CFE bit in the RSCFDnCFDCFCCK register is set to 1 and the CFFLL flag in the RSCFDnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer is not full).
- Gateway mode (CFM[1:0] value in the RSCFDnCFDCFCCK register is 10_B):
Setting prohibited

17.4.8.4 RSCFDnCFDCFIDk — Transmit/receive FIFO Buffer Access ID Register k (k = 0 to 8)

Access: RSCFDnCFDCFIDk register can be read/written in 32-bit units
RSCFDnCFDCFIDkL, RSCFDnCFDCFIDkH registers can be read/written in 16-bit units
RSCFDnCFDCFIDkLL, RSCFDnCFDCFIDkLH, RSCFDnCFDCFIDkHL, RSCFDnCFDCFIDkHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFIDk: $\langle \text{RSCFDn_base} \rangle + 3400_{\text{H}} + (80_{\text{H}} \times k)$
RSCFDnCFDCFIDkL: $\langle \text{RSCFDn_base} \rangle + 3400_{\text{H}} + (80_{\text{H}} \times k)$,
RSCFDnCFDCFIDkH: $\langle \text{RSCFDn_base} \rangle + 3402_{\text{H}} + (80_{\text{H}} \times k)$
RSCFDnCFDCFIDkLL: $\langle \text{RSCFDn_base} \rangle + 3400_{\text{H}} + (80_{\text{H}} \times k)$,
RSCFDnCFDCFIDkLH: $\langle \text{RSCFDn_base} \rangle + 3401_{\text{H}} + (80_{\text{H}} \times k)$,
RSCFDnCFDCFIDkHL: $\langle \text{RSCFDn_base} \rangle + 3402_{\text{H}} + (80_{\text{H}} \times k)$,
RSCFDnCFDCFIDkHH: $\langle \text{RSCFDn_base} \rangle + 3403_{\text{H}} + (80_{\text{H}} \times k)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-------|-----|-----|-------|-----|-----|-------------|-----|-----|-----|-----|-----|-----|
| | CFIDE | | | CFRTR | | | THLEN | | | CFID[28:16] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.135 RSCFDnCFDCFIDk Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 | CFIDE | Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | CFRTR | Transmit/Receive FIFO Buffer RTR/RRS <ul style="list-style-type: none"> When the CFM[1:0] value is 01_B (transmit mode) <ul style="list-style-type: none"> When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame When the transmit message is a CAN FD frame Write 0 to this bit. When the CFM[1:0] value is 00_B (receive mode) <ul style="list-style-type: none"> When the received message is a classical CAN frame 0: Data frame 1: Remote frame When the received message is a CAN FD frame The RRS bit value of the received message can be read. |
| 29 | THLEN | Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01 _B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer. |
| 28 to 0 | CFID[28:0] | Transmit/Receive FIFO Buffer ID Data <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0. |

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). This RSCFDnCFDCFIDk register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

If the the received message is a classical CAN frame, this bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. If the received message is a CAN FD frame, this bit indicates the RRS bit value of the received message.

When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFFDF bit in the RSCFDnCFDCFFDCSTSk register is 1 (CAN FD frame), set this bit to 0.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, these bits are used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

17.4.8.5 RSCFDnCFDCFPTRk — Transmit/receive FIFO Buffer Access Pointer Register k (k = 0 to 8)

Access: RSCFDnCFDCFPTRk register can be read/written in 32-bit units
 RSCFDnCFDCFPTRkL, RSCFDnCFDCFPTRkH registers can be read/written in 16-bit units
 RSCFDnCFDCFPTRkLL, RSCFDnCFDCFPTRkLH, RSCFDnCFDCFPTRkHL, RSCFDnCFDCFPTRkHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFPTRk: $\langle \text{RSCFDn_base} \rangle + 3404_{\text{H}} + (80_{\text{H}} \times k)$
 RSCFDnCFDCFPTRkL: $\langle \text{RSCFDn_base} \rangle + 3404_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFPTRkH: $\langle \text{RSCFDn_base} \rangle + 3406_{\text{H}} + (80_{\text{H}} \times k)$
 RSCFDnCFDCFPTRkLL: $\langle \text{RSCFDn_base} \rangle + 3404_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFPTRkLH: $\langle \text{RSCFDn_base} \rangle + 3405_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFPTRkHL: $\langle \text{RSCFDn_base} \rangle + 3406_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFPTRkHH: $\langle \text{RSCFDn_base} \rangle + 3407_{\text{H}} + (80_{\text{H}} \times k)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | CFDLC[3:0] | | | | CFPTR[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.136 RSCFDnCFDCFPTRk Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|-------------|---|-----|---------------------|---------------|-----|---------------------|--------------|---|---|---|---|--------------|--|---|---|---|---|-------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|--|---|---|---|---|--------------|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|---|---|---|---|--|---------------|
| 31 to 28 | CFDLC[3:0] | Transmit/Receive FIFO Buffer DLC Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>b31</th> <th>b30</th> <th>b29</th> <th>b28</th> <th>Classical CAN Frame</th> <th>CAN FD Frame</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1 data byte</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>2 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>3 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>4 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>5 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>6 data bytes</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>7 data bytes</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>8 data bytes</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>8 data bytes</td> <td>12 data bytes</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td></td> <td>16 data bytes</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td></td> <td>20 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td></td> <td>24 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td></td> <td>32 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td></td> <td>48 data bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td></td> <td>64 data bytes</td> </tr> </tbody> </table> | b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | 0 | 0 | 0 | 0 | 0 data bytes | | 0 | 0 | 0 | 1 | 1 data byte | | 0 | 0 | 1 | 0 | 2 data bytes | | 0 | 0 | 1 | 1 | 3 data bytes | | 0 | 1 | 0 | 0 | 4 data bytes | | 0 | 1 | 0 | 1 | 5 data bytes | | 0 | 1 | 1 | 0 | 6 data bytes | | 0 | 1 | 1 | 1 | 7 data bytes | | 1 | 0 | 0 | 0 | 8 data bytes | | 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | 1 | 0 | 1 | 0 | | 16 data bytes | 1 | 0 | 1 | 1 | | 20 data bytes | 1 | 1 | 0 | 0 | | 24 data bytes | 1 | 1 | 0 | 1 | | 32 data bytes | 1 | 1 | 1 | 0 | | 48 data bytes | 1 | 1 | 1 | 1 | | 64 data bytes |
| b31 | b30 | b29 | b28 | Classical CAN Frame | CAN FD Frame | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1 data byte | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 2 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 3 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 4 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 5 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 6 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 7 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 8 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 1 | 8 data bytes | 12 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | | 16 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | | 20 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | | 24 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 1 | | 32 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 0 | | 48 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | | 64 data bytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 27 to 16 | CFPTR[11:0] | Transmit/Receive FIFO Buffer Label Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <ul style="list-style-type: none"> When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 17.136 RSCFDnCFDCFPTRk Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 15 to 0 | CFTS[15:0] | Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is 00 _B (receive mode). The timestamp value of the received message can be read. |

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFDLC[3:0] bits are set to 1001_B or more while the CFFDF bit in the RSCFDnCFDCFFDCSTSk register is 0 (classical CAN frame), 8-byte data is transmitted actually. When the CFFDF bit is 1 (CAN FD frame), the settable value range varies depending on the settings of the TMME bit in the RSCFDnCFDCmFDCFG register and the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register.

- When TMME bit = 0 (transmit buffer merge mode disabled):
A value of 0000_B to 1111_B is settable. If the specified data length exceeds the payload storage size specified by the CFPLS[2:0] bits, excessive payloads are padded by CCH.
- When TMME bit = 1 (transmit buffer merge mode enabled):
Set the data length within the payload storage size specified by the CFPLS[2:0] bits.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

17.4.8.6 RSCFDnCFDCFFDCSTSk — Transmit/Receive FIFO CAN FD Configuration/ Status Register k (k = 0 to 8)

Access: RSCFDnCFDCFFDCSTSk register can be read/written in 32-bit units
 RSCFDnCFDCFFDCSTSkL, RSCFDnCFDCFFDCSTSkH registers can be read/written in 16-bit units
 RSCFDnCFDCFFDCSTSkLL, RSCFDnCFDCFFDCSTSkLH, RSCFDnCFDCFFDCSTSkHL,
 RSCFDnCFDCFFDCSTSkHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFFDCSTSk: $\langle \text{RSCFDn_base} \rangle + 3408_{\text{H}} + (80_{\text{H}} \times k)$
 RSCFDnCFDCFFDCSTSkL: $\langle \text{RSCFDn_base} \rangle + 3408_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFFDCSTSkH: $\langle \text{RSCFDn_base} \rangle + 340A_{\text{H}} + (80_{\text{H}} \times k)$
 RSCFDnCFDCFFDCSTSkLL: $\langle \text{RSCFDn_base} \rangle + 3408_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFFDCSTSkLH: $\langle \text{RSCFDn_base} \rangle + 3409_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFFDCSTSkHL: $\langle \text{RSCFDn_base} \rangle + 340A_{\text{H}} + (80_{\text{H}} \times k)$,
 RSCFDnCFDCFFDCSTSkHH: $\langle \text{RSCFDn_base} \rangle + 340B_{\text{H}} + (80_{\text{H}} \times k)$

Initial value: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CFDF | CFBRS | CFESI |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 17.137 RSCFDnCFDCFFDCSTSk Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 2 | CFDF | FD 0: Classical CAN frame 1: CAN FD frame |
| 1 | CFBRS | BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes. |
| 0 | CFBRS | ESI 0: Error active node 1: Error passive node |

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode). This register can be read only when the CFM[1:0] value is 00_B (receive mode). Do not read or write this register when the CFM[1:0] value is 10_B (gateway mode).

CFDF Bit

When the CFM[1:0] value is 00_B, this bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the transmit/receive FIFO buffer. When the CFM[1:0] value is 01_B, this bit is used to set the FD format of the message to be transmitted from the transmit/receive FIFO buffer.

CFBRS Bit

When the CFM[1:0] value is 00_B, if the CFFDF bit is set to 1, this bit indicates the BRS bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, this bit is always read as 0.

When the CFM[1:0] value is 01_B, if the CFFDF bit is set to 1, this bit is used to set the BRS bit value of the message to be transmitted from the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, write 0 to this bit.

CFESI Bit

When the CFM[1:0] value is 00_B, if the CFFDF bit is set to 1, this bit indicates the ESI bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is set to 0, this bit is always read as 0.

When the CFM[1:0] value is 01_B, if the CFFDF bit is set to 1, this bit is used to set the ESI bit value of the message to be transmitted from the transmit/receive FIFO buffer. The set value is transmitted when the ESIC bit in the RSCFDnCFDCmFDCFG register is set to 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the CFFDF bit is set to 0, write 0 to this bit.

17.4.8.7 RSCFDnCFDCFDf_d_k — Transmit/receive FIFO Buffer Access Data Field d Register k (d = 0 to 15, k = 0 to 8)

Access: RSCFDnCFDCFDf_d_k register can be read/written in 32-bit units
 RSCFDnCFDCFDf_d_kL, RSCFDnCFDCFDf_d_kH registers can be read/written in 16-bit units
 RSCFDnCFDCFDf_d_kLL, RSCFDnCFDCFDf_d_kLH, RSCFDnCFDCFDf_d_kHL, RSCFDnCFDCFDf_d_kHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFDf_d_k: $\langle \text{RSCFDn_base} \rangle + 340C_H + (04_H \times d) + (80_H \times k)$
 RSCFDnCFDCFDf_d_kL: $\langle \text{RSCFDn_base} \rangle + 340C_H + (04_H \times d) + (80_H \times k)$,
 RSCFDnCFDCFDf_d_kH: $\langle \text{RSCFDn_base} \rangle + 340E_H + (04_H \times d) + (80_H \times k)$
 RSCFDnCFDCFDf_d_kLL: $\langle \text{RSCFDn_base} \rangle + 340C_H + (04_H \times d) + (80_H \times k)$,
 RSCFDnCFDCFDf_d_kLH: $\langle \text{RSCFDn_base} \rangle + 340D_H + (04_H \times d) + (80_H \times k)$,
 RSCFDnCFDCFDf_d_kHL: $\langle \text{RSCFDn_base} \rangle + 340E_H + (04_H \times d) + (80_H \times k)$,
 RSCFDnCFDCFDf_d_kHH: $\langle \text{RSCFDn_base} \rangle + 340F_H + (04_H \times d) + (80_H \times k)$

Initial value: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|---------------------|-----|-----|-----|-----|-----|-----|-----|---------------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CFDB4 × d + 3 [7:0] | | | | | | | | CFDB4 × d + 2 [7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CFDB4 × d + 1 [7:0] | | | | | | | | CFDB4 × d + 0 [7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.138 RSCFDnCFDCFDf_d_k Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 24 | CFDB4 × d + 3 [7:0] | Transmit/Receive FIFO Buffer Data Byte 4 × d + 3 Transmit/Receive FIFO Buffer Data Byte 4 × d + 2 |
| 23 to 16 | CFDB4 × d + 2 [7:0] | Transmit/Receive FIFO Buffer Data Byte 4 × d + 1 Transmit/Receive FIFO Buffer Data Byte 4 × d + 0 |
| 15 to 8 | CFDB4 × d + 1 [7:0] | <ul style="list-style-type: none"> When CFM[1:0] value is 01B (transmit mode): Set the transmit/receive FIFO buffer data. |
| 7 to 0 | CFDB4 × d + 0 [7:0] | <ul style="list-style-type: none"> When CFM[1:0] value is 00B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read. |

This register can be written only when the CFM[1:0] value in the RSCFDnCFDCFCCK register is 01_B (transmit mode).

This register can be read only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCFDnCFDCFPTRk register is smaller than the payload storage size of the transmit/receive FIFO buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the transmit/receive FIFO buffer by the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register. Do not read or write the RSCFDnCFDCFDf_d_k register corresponding to an area that exceeds the specified size.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

17.4.9 Details of FIFO Status-related Registers

17.4.9.1 RSCFDnCFDFESTS — FIFO Empty Status Register

Access: RSCFDnCFDFESTS register can be read only in 32-bit units
 RSCFDnCFDFESTSL, RSCFDnCFDFESTSH registers can be read only in 16-bit units
 RSCFDnCFDFESTSLL, RSCFDnCFDFESTSLH, RSCFDnCFDFESTSHL, RSCFDnCFDFESTSHH registers can be read only in 8-bit units

Address: RSCFDnCFDFESTS: <RSCFDn_base> + 0238_H
 RSCFDnCFDFESTSL: <RSCFDn_base> + 0238_H, RSCFDnCFDFESTSH: <RSCFDn_base> + 023A_H
 RSCFDnCFDFESTSLL: <RSCFDn_base> + 0238_H, RSCFDnCFDFESTSLH: <RSCFDn_base> + 0239_H,
 RSCFDnCFDFESTSHL: <RSCFDn_base> + 023A_H, RSCFDnCFDFESTSHH: <RSCFDn_base> + 023B_H

Value after reset: 03FF FFFF_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CF8EMP |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CF7EMP | CF6EMP | CF5EMP | CF4EMP | CF3EMP | CF2EMP | CF1EMP | CF0EMP | RF7EMP | RF6EMP | RF5EMP | RF4EMP | RF3EMP | RF2EMP | RF1EMP | RF0EMP |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.139 RSCFDnCFDFESTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 17 | Reserved | These bits are read as the value after reset. |
| 16 | CF8EMP | Transmit/Receive FIFO Buffer Empty Status Flag |
| 15 | CF7EMP | 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. |
| 14 | CF6EMP | (k = 0 to 8) |
| 13 | CF5EMP | |
| 12 | CF4EMP | |
| 11 | CF3EMP | |
| 10 | CF2EMP | |
| 9 | CF1EMP | |
| 8 | CF0EMP | |
| 7 | RF7EMP | Receive FIFO Buffer Empty Status Flag |
| 6 | RF6EMP | 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message. |
| 5 | RF5EMP | (x = 0 to 7) |
| 4 | RF4EMP | |
| 3 | RF3EMP | |
| 2 | RF2EMP | |
| 1 | RF1EMP | |
| 0 | RF0EMP | |

The RSCFDnCFDFESTS register is set to 03FF FFFF_H in global reset mode.

CFkEMP Flag (k = 0 to 8)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCFDnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCFDnCFDRFSTx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

17.4.9.2 RSCFDnCFDFFSTS — FIFO Full Status Register

Access: RSCFDnCFDFFSTS register can be read only in 32-bit units
 RSCFDnCFDFFSTSL, RSCFDnCFDFFSTSH registers can be read only in 16-bit units
 RSCFDnCFDFFSTSL, RSCFDnCFDFFSTSLH, RSCFDnCFDFFSTSHL, RSCFDnCFDFFSTSHH registers can be read only in 8-bit units

Address: RSCFDnCFDFFSTS: <RSCFDn_base> + 023C_H
 RSCFDnCFDFFSTSL: <RSCFDn_base> + 023C_H, RSCFDnCFDFFSTSH: <RSCFDn_base> + 023E_H
 RSCFDnCFDFFSTSL: <RSCFDn_base> + 023C_H, RSCFDnCFDFFSTSLH: <RSCFDn_base> + 023D_H,
 RSCFDnCFDFFSTSHL: <RSCFDn_base> + 023E_H, RSCFDnCFDFFSTSHH: <RSCFDn_base> + 023F_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CF8FLL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | CF7FLL | CF6FLL | CF5FLL | CF4FLL | CF3FLL | CF2FLL | CF1FLL | CF0FLL | RF7FLL | RF6FLL | RF5FLL | RF4FLL | RF3FLL | RF2FLL | RF1FLL | RF0FLL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.140 RSCFDnCFDFFSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 17 | Reserved | These bits are read as the value after reset. |
| 16 | CF8FLL | Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. |
| 15 | CF7FLL | |
| 14 | CF6FLL | (k = 0 to 8) |
| 13 | CF5FLL | |
| 12 | CF4FLL | |
| 11 | CF3FLL | |
| 10 | CF2FLL | |
| 9 | CF1FLL | |
| 8 | CF0FLL | |
| 7 | RF7FLL | Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. |
| 6 | RF6FLL | (x = 0 to 7) |
| 5 | RF5FLL | |
| 4 | RF4FLL | |
| 3 | RF3FLL | |
| 2 | RF2FLL | |
| 1 | RF1FLL | |
| 0 | RF0FLL | |

The RSCFDnCFDFFSTS register is cleared to 0000 0000_H in global reset mode.

CFkFLL Flag (k = 0 to 8)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCFDnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).

When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLl Flag (x = 0 to 7)

The RFxFLl flag is set to 1 when the RFFLL flag in the RSCFDnCFDRFSTsx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLl flag is cleared to 0.

17.4.9.3 RSCFDnCFDFMSTS — FIFO Message Lost Status Register

Access: RSCFDnCFDFMSTS register can be read only in 32-bit units
 RSCFDnCFDFMSTSL, RSCFDnCFDFMSTSH registers can be read only in 16-bit units
 RSCFDnCFDFMSTSL, RSCFDnCFDFMSTSLH, RSCFDnCFDFMSTSHL, RSCFDnCFDFMSTSHH registers can be read only in 8-bit units

Address: RSCFDnCFDFMSTS: <RSCFDn_base> + 0240_H
 RSCFDnCFDFMSTSL: <RSCFDn_base> + 0240_H, RSCFDnCFDFMSTSH: <RSCFDn_base> + 0242_H
 RSCFDnCFDFMSTSL: <RSCFDn_base> + 0240_H, RSCFDnCFDFMSTSLH: <RSCFDn_base> + 0241_H,
 RSCFDnCFDFMSTSHL: <RSCFDn_base> + 0242_H, RSCFDnCFDFMSTSHH: <RSCFDn_base> + 0243_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CF8MLT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | CF7MLT | CF6MLT | CF5MLT | CF4MLT | CF3MLT | CF2MLT | CF1MLT | CF0MLT | RF7MLT | RF6MLT | RF5MLT | RF4MLT | RF3MLT | RF2MLT | RF1MLT | RF0MLT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.141 RSCFDnCFDFMSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 17 | Reserved | These bits are read as the value after reset. |
| 16 | CF8MLT | Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. |
| 15 | CF7MLT | |
| 14 | CF6MLT | (k = 0 to 8) |
| 13 | CF5MLT | |
| 12 | CF4MLT | |
| 11 | CF3MLT | |
| 10 | CF2MLT | |
| 9 | CF1MLT | |
| 8 | CF0MLT | |
| 7 | RF7MLT | Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. |
| 6 | RF6MLT | |
| 5 | RF5MLT | (x = 0 to 7) |
| 4 | RF4MLT | |
| 3 | RF3MLT | |
| 2 | RF2MLT | |
| 1 | RF1MLT | |
| 0 | RF0MLT | |

The RSCFDnCFDFMSTS register is cleared to 0000 0000_H in global reset mode.

CFkMLT Flag (k = 0 to 8)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCFDnCFDFMSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCFDnCFDRFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

17.4.9.4 RSCFDnCFDRFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: RSCFDnCFDRFISTS register can be read only in 32-bit units
 RSCFDnCFDRFISTS_{SL}, RSCFDnCFDRFISTS_{SH} registers can be read only in 16-bit units
 RSCFDnCFDRFISTS_{SL}_L, RSCFDnCFDRFISTS_{SL}_{LH}, RSCFDnCFDRFISTS_{SH}_L, RSCFDnCFDRFISTS_{SH}_H
 registers can be read only in 8-bit units

Address: RSCFDnCFDRFISTS: <RSCFDn_base> + 0244_H
 RSCFDnCFDRFISTS_{SL}: <RSCFDn_base> + 0244_H, RSCFDnCFDRFISTS_{SH}: <RSCFDn_base> + 0246_H
 RSCFDnCFDRFISTS_{SL}_L: <RSCFDn_base> + 0244_H, RSCFDnCFDRFISTS_{SL}_{LH}: <RSCFDn_base> + 0245_H,
 RSCFDnCFDRFISTS_{SH}_L: <RSCFDn_base> + 0246_H, RSCFDnCFDRFISTS_{SH}_H: <RSCFDn_base> + 0247_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|-------|-------|-------|-------|-------|-------|-------|-------|
| | — | — | — | — | — | — | — | — | RF7IF | RF6IF | RF5IF | RF4IF | RF3IF | RF2IF | RF1IF | RF0IF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.142 RSCFDnCFDRFISTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 8 | Reserved | These bits are read as the value after reset. |
| 7 | RF7IF | Receive FIFO Buffer Interrupt Request Status Flag |
| 6 | RF6IF | 0: No receive FIFO buffer x interrupt request is present. |
| 5 | RF5IF | 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7) |
| 4 | RF4IF | |
| 3 | RF3IF | |
| 2 | RF2IF | |
| 1 | RF1IF | |
| 0 | RF0IF | |

The RSCFDnCFDRFISTS register is cleared to 0000 0000_H in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RSCFDnCFDRFISTS_x register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

17.4.9.5 RSCFDnCFDCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RSCFDnCFDCFRISTS register can be read only in 32-bit units
RSCFDnCFDCFRISTSL, RSCFDnCFDCFRISTSH registers can be read only in 16-bit units
RSCFDnCFDCFRISTSLL, RSCFDnCFDCFRISTSLH, RSCFDnCFDCFRISTSHL, RSCFDnCFDCFRISTSHH registers can be read only in 8-bit units

Address: RSCFDnCFDCFRISTS: <RSCFDn_base> + 0248_H
RSCFDnCFDCFRISTSL: <RSCFDn_base> + 0248_H, RSCFDnCFDCFRISTSH: <RSCFDn_base> + 024A_H
RSCFDnCFDCFRISTSLL: <RSCFDn_base> + 0248_H, RSCFDnCFDCFRISTSLH: <RSCFDn_base> + 0249_H,
RSCFDnCFDCFRISTSHL: <RSCFDn_base> + 024A_H, RSCFDnCFDCFRISTSHH: <RSCFDn_base> + 024B_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | CF8RXI | CF7RXI | CF6RXI | CF5RXI | CF4RXI | CF3RXI | CF2RXI | CF1RXI | CF0RXI |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.143 RSCFDnCFDCFRISTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 9 | Reserved | These bits are read as the value after reset. |
| 8 | CF8RXIF | Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 8) |
| 7 | CF7RXIF | |
| 6 | CF6RXIF | |
| 5 | CF5RXIF | |
| 4 | CF4RXIF | |
| 3 | CF3RXIF | |
| 2 | CF2RXIF | |
| 1 | CF1RXIF | |
| 0 | CF0RXIF | |

The RSCFDnCFDCFRISTS register is cleared to 0000 0000_H in global reset mode.

CFkRXIF Flag (k = 0 to 8)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCFDnCFDCFRISTS_k register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

17.4.9.6 RSCFDnCFDCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RSCFDnCFDCFTISTS register can be read only in 32-bit units
RSCFDnCFDCFTISTS_{SL}, RSCFDnCFDCFTISTS_{SH} registers can be read only in 16-bit units
RSCFDnCFDCFTISTS_{SLL}, RSCFDnCFDCFTISTS_{SLH}, RSCFDnCFDCFTISTS_{SHL}, RSCFDnCFDCFTISTS_{SHH} registers can be read only in 8-bit units

Address: RSCFDnCFDCFTISTS: <RSCFDn_base> + 024C_H
RSCFDnCFDCFTISTS_{SL}: <RSCFDn_base> + 024C_H, RSCFDnCFDCFTISTS_{SH}: <RSCFDn_base> + 024E_H
RSCFDnCFDCFTISTS_{SLL}: <RSCFDn_base> + 024C_H, RSCFDnCFDCFTISTS_{SLH}: <RSCFDn_base> + 024D_H,
RSCFDnCFDCFTISTS_{SHL}: <RSCFDn_base> + 024E_H, RSCFDnCFDCFTISTS_{SHH}: <RSCFDn_base> + 024F_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | CF8TXI | CF7TXI | CF6TXI | CF5TXI | CF4TXI | CF3TXI | CF2TXI | CF1TXI | CF0TXI |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.144 RSCFDnCFDCFTISTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 9 | Reserved | These bits are read as the value after reset. |
| 8 | CF8TXIF | Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 8) |
| 7 | CF7TXIF | |
| 6 | CF6TXIF | |
| 5 | CF5TXIF | |
| 4 | CF4TXIF | |
| 3 | CF3TXIF | |
| 2 | CF2TXIF | |
| 1 | CF1TXIF | |
| 0 | CF0TXIF | |

The RSCFDnCFDCFTISTS register is cleared to 0000 0000_H in global reset mode.

CFkTXIF Flag (k = 0 to 8)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCFDnCFDCFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

17.4.10 Details of FIFO DMA-Related Registers

17.4.10.1 RSCFDnCFDCDTCT — DMA Enable Register

Access: RSCFDnCFDCDTCT register can be read/written in 32-bit units
 RSCFDnCFDCDTCTL, RSCFDnCFDCDTCTH registers can be read/written in 16-bit units
 RSCFDnCFDCDTCTL, RSCFDnCFDCDTCTLH, RSCFDnCFDCDTCTHL, RSCFDnCFDCDTCTHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCDTCT: <RSCFDn_base> + 0490_H
 RSCFDnCFDCDTCTL: <RSCFDn_base> + 0490_H
 RSCFDnCFDCDTCTH: <RSCFDn_base> + 0492_H
 RSCFDnCFDCDTCTL: <RSCFDn_base> + 0490_H,
 RSCFDnCFDCDTCTLH: <RSCFDn_base> + 0491_H,
 RSCFDnCFDCDTCTHL: <RSCFDn_base> + 0492_H,
 RSCFDnCFDCDTCTHH: <RSCFDn_base> + 0493_H

Initial value: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | CFDMA E2 | CFDMA E1 | CFDMA E0 | RFDMA E7 | RFDMA E6 | RFDMA E5 | RFDMA E4 | RFDMA E3 | RFDMA E2 | RFDMA E1 | RFDMA E0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.145 RSCFDnCFDCDTCT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 11 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 10 | CFDMAE2 | Transmit/Receive FIFO Buffer 6 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 6 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 6 is enabled. |
| 9 | CFDMAE1 | Transmit/Receive FIFO Buffer 3 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 3 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 3 is enabled. |
| 8 | CFDMAE0 | Transmit/Receive FIFO Buffer 0 DMA Enable 0: A DMA transfer request of transmit/receive FIFO buffer 0 is disabled. 1: A DMA transfer request of transmit/receive FIFO buffer 0 is enabled. |
| 7 | RFDMAE7 | Receive FIFO Buffer x DMA Enable 0: A DMA transfer request of receive FIFO buffer x is disabled. 1: A DMA transfer request of receive FIFO buffer x is enabled. (x = 0 to 7) |
| 6 | RFDMAE6 | |
| 5 | RFDMAE5 | |
| 4 | RFDMAE4 | |
| 3 | RFDMAE3 | |
| 2 | RFDMAE2 | |
| 1 | RFDMAE1 | |
| 0 | RFDMAE0 | |

Modify the RSCFDnCFDCDTCT register in global operating mode or global test mode.

CFDMAEm Bit

This bit is used to enable DMA transfer for transmit/receive FIFO buffer $3 \times m$ (the first transmit/receive FIFO buffer allocated to channel m). DMA transfer is enabled only for transmit/receive FIFO buffers for which the CFM[1:0] bits in the RSCFDnCFDCFCCK register is set to 00_B (receive mode). Set this bit to 0 when the CFM[1:0] value is 01_B (transmit mode) or 10_B (gateway mode).

RFDMAEx Bit

This bit is used to enable DMA transfer for receive FIFO buffer x.

17.4.10.2 RSCFDnCFDCDTSTS — DMA Status Register

Access: RSCFDnCFDCDTSTS register can be read only in 32-bit units
 RSCFDnCFDCDTSTSL, RSCFDnCFDCDTSTSH registers can be read only in 16-bit units
 RSCFDnCFDCDTSTSLL, RSCFDnCFDCDTSTSLH, RSCFDnCFDCDTSTSHL, RSCFDnCFDCDTSTSHH registers can be read only in 8-bit units

Address: RSCFDnCFDCDTSTS: <RSCFDn_base> + 0494_H
 RSCFDnCFDCDTSTSL: <RSCFDn_base> + 0494_H
 RSCFDnCFDCDTSTSH: <RSCFDn_base> + 0496_H
 RSCFDnCFDCDTSTSLL: <RSCFDn_base> + 0494_H,
 RSCFDnCFDCDTSTSLH: <RSCFDn_base> + 0495_H,
 RSCFDnCFDCDTSTSHL: <RSCFDn_base> + 0496_H,
 RSCFDnCFDCDTSTSHH: <RSCFDn_base> + 0497_H

Initial value: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | CFDMA STS2 | CFDMA STS1 | CFDMA STS0 | RFDMA STS7 | RFDMA STS6 | RFDMA STS5 | RFDMA STS4 | RFDMA STS3 | RFDMA STS2 | RFDMA STS1 | RFDMA STS0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.146 RSCFDnCFDCDTSTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 11 | Reserved | These bits are read as the value after reset. |
| 10 | CFDMASTS2 | Transmit/Receive FIFO Buffer 6 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 6 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 6 is in progress. |
| 9 | CFDMASTS1 | Transmit/Receive FIFO Buffer 3 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 3 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 3 is in progress. |
| 8 | CFDMASTS0 | Transmit/Receive FIFO Buffer 0 DMA Status 0: DMA transfer of transmit/receive FIFO buffer 0 is not in progress. 1: DMA transfer of transmit/receive FIFO buffer 0 is in progress. |
| 7 | RFDMASTS7 | Receive FIFO Buffer x DMA Status 0: DMA transfer of receive FIFO buffer x is not in progress. 1: DMA transfer of receive FIFO buffer x is in progress. (x = 0 to 7) |
| 6 | RFDMASTS6 | |
| 5 | RFDMASTS5 | |
| 4 | RFDMASTS4 | |
| 3 | RFDMASTS3 | |
| 2 | RFDMASTS2 | |
| 1 | RFDMASTS1 | |
| 0 | RFDMASTS0 | |

CFDMASTSm Bit

When DMA transfer is enabled (CFDMAEm bit in the RSCFDnCFDCDTCT register is 1) for the transmit/receive FIFO buffer $3 \times m$ (the first transmit/receive FIFO buffer allocated to channel m) while the transmit/receive FIFO buffer contains one of more messages, the CFDMASTSm bit is set to 1 indicating that DMA transfer is in progress.

When all messages in the transmit/receive FIFO buffer have been transferred or DMA transfer is disabled (CFDMAEm bit is 0), the CFDMASTSm bit is cleared to 0 indicating that DMA transfer has been completed. If the CFDMAEm bit is set to 0 during DMA transfer, the CFDMASTSm bit is cleared to 0 after the ongoing DMA transfer has been completed (when the message that is being transferred has been transferred to the last byte in the payload storage area).

These bits are cleared to 0 in global reset mode.

RFDMASTSm Bit

When DMA transfer is enabled (corresponding RFDMAEx bit in the RSCFDnCFDCDTCT register is 1) for the receive FIFO buffer x and the receive FIFO buffer contains one of more messages, the RFDMASTSm bit is set to 1 indicating that DMA transfer is in progress.

When all messages in the receive FIFO buffer x have been transferred or DMA transfer is disabled (RFDMAEx bit = 0), the RFDMASTSm bit is cleared to 0 indicating that DMA transfer has been completed. If the RFDMAEx bit is set to 0 during DMA transfer, the RFDMASTSm bit is cleared to 0 after the ongoing DMA transfer has been completed (when the message that is being transferred has been transferred to the last byte in the payload storage area)

These bits are cleared to 0 in global reset mode.

17.4.11 Details of Transmit Buffer-related Registers

17.4.11.1 RSCFDnCFDTMCp — Transmit Buffer Control Register p (p = 0 to 47)

Access: RSCFDnCFDTMCp register can be read/written in 8-bit units

Address: RSCFDnCFDTMCp: <RSCFDn_base> + 0250_H + (01_H × p)

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|------|-------------------|-------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | TMOM | TMTAR | TMTR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W* ¹ | R/W* ¹ |

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 17.147 RSCFDnCFDTMCp Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 3 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 2 | TMOM | One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled. |
| 1 | TMTAR | Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested. |
| 0 | TMTR | Transmit Request 0: Transmission is not requested. 1: Transmission is requested. |

When the RSCFDnCFDTMCp register meets any of the following conditions, set it to 00_H.

- The RSCFDnCFDTMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCFDnCFDCFCCK register (p = m × 16 + the value of CFTML[3:0] bits).
- The RSCFDnCFDTMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCFDnCFDCTXQCCm (m = 0 to 2) register (p = (m × 16 + 15) to (m × 16 + 15 + the value of TXQDC[3:0] bits)).
- RSCFDnCFDTMCp register (p = (m × 16) + 1, (m × 16) + 2) corresponding to the transmit buffer allocated as a payload storage area when the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode)

All of the bits in the RSCFDnCFDTMCp register are cleared to 0 in channel reset mode. Modify the RSCFDnCFDTMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. If transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCFDnCFDTMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RSCFDnCFDTMSTSp register is 00_B.

17.4.11.2 RSCFDnCFDTMSTSp — Transmit Buffer Status Register p (p = 0 to 47)

Access: RSCFDnCFDTMSTSp register can be read/written in 8-bit units

Address: RSCFDnCFDTMSTSp: <RSCFDn_base> + 02D0_H + (01_H × p)

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|--------|-------|------------|-----|--------|
| | — | — | — | TMTARM | TMTRM | TMTRF[1:0] | | TMTSTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R |

Table 17.148 RSCFDnCFDTMSTSp Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 7 to 5 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 4 | TMTARM | Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present. |
| 3 | TMTRM | Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present. |
| 2, 1 | TMTRF[1:0] | Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request). |
| 0 | TMTSTS | Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress. |

All of the bits in the RSCFDnCFDTMSTSp register are cleared to 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCFDnCFDTMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCFDnCFDTMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCFDnCFDTMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCFDnCFDTMCp register is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCFDnCFDTMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCFDnCFDTMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

17.4.11.3 RSCFDnCFDTMIDp — Transmit Buffer ID Register p (p = 0 to 47)

Access: RSCFDnCFDTMIDp register can be read/written in 32-bit units
 RSCFDnCFDTMIDpL, RSCFDnCFDTMIDpH registers can be read/written in 16-bit units
 RSCFDnCFDTMIDpLL, RSCFDnCFDTMIDpLH, RSCFDnCFDTMIDpHL, RSCFDnCFDTMIDpHH registers
 can be read/written in 8-bit units

Address: RSCFDnCFDTMIDp: <RSCFDn_base> + 4000_H + (20_H × p)
 RSCFDnCFDTMIDpL: <RSCFDn_base> + 4000_H + (20_H × p),
 RSCFDnCFDTMIDpH: <RSCFDn_base> + 4002_H + (20_H × p)
 RSCFDnCFDTMIDpLL: <RSCFDn_base> + 4000_H + (20_H × p),
 RSCFDnCFDTMIDpLH: <RSCFDn_base> + 4001_H + (20_H × p),
 RSCFDnCFDTMIDpHL: <RSCFDn_base> + 4002_H + (20_H × p),
 RSCFDnCFDTMIDpHH: <RSCFDn_base> + 4003_H + (20_H × p)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-------|-------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | TMIDE | TMRTR | THLEN | TMID[28:16] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.149 RSCFDnCFDTMIDp Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 | TMIDE | Transmit Buffer IDE 0: Standard ID 1: Extended ID |
| 30 | TMRTR | Transmit Buffer RTR/RRS <ul style="list-style-type: none"> When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame When the transmit message is a CAN FD frame Write 0 to this bit. |
| 29 | THLEN | Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer. |
| 28 to 0 | TMID[28:0] | Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. |

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

Set this bit to 0 when the TMFDF bit in the RSCFDnCFDTMFDCTR_p register is 1 (CAN FD frame).

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

17.4.11.4 RSCFDnCFDTMPTRp — Transmit Buffer Pointer Register p (p = 0 to 47)

Access: RSCFDnCFDTMPTRp register can be read/written in 32-bit units
 RSCFDnCFDTMPTRpL, RSCFDnCFDTMPTRpH registers can be read/written in 16-bit units
 RSCFDnCFDTMPTRpLL, RSCFDnCFDTMPTRpLH, RSCFDnCFDTMPTRpHL, RSCFDnCFDTMPTRpHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTMPTRp: <RSCFDn_base> + 4004_H + (20_H × p)
 RSCFDnCFDTMPTRpL: <RSCFDn_base> + 4004_H + (20_H × p),
 RSCFDnCFDTMPTRpH: <RSCFDn_base> + 4006_H + (20_H × p)
 RSCFDnCFDTMPTRpLL: <RSCFDn_base> + 4004_H + (20_H × p),
 RSCFDnCFDTMPTRpLH: <RSCFDn_base> + 4005_H + (20_H × p),
 RSCFDnCFDTMPTRpHL: <RSCFDn_base> + 4006_H + (20_H × p),
 RSCFDnCFDTMPTRpHH: <RSCFDn_base> + 4007_H + (20_H × p)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|----|----|----|----|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TMDLC[3:0] | | | | — | — | — | — | TMPTR[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.150 RSCFDnCFDTMPTRp Register Contents

| Bit Position | Bit Name | Function | | | |
|--------------|------------|--|-----------------|---------------------|---------------|
| 31 to 28 | TMDLC[3:0] | Transmit Buffer DLC Data | | | |
| | | | b31 b30 b29 b28 | Classical CAN Frame | CAN FD Frame |
| | | | 0 0 0 0 | 0 data bytes | |
| | | | 0 0 0 1 | 1 data byte | |
| | | | 0 0 1 0 | 2 data bytes | |
| | | | 0 0 1 1 | 3 data bytes | |
| | | | 0 1 0 0 | 4 data bytes | |
| | | | 0 1 0 1 | 5 data bytes | |
| | | | 0 1 1 0 | 6 data bytes | |
| | | | 0 1 1 1 | 7 data bytes | |
| | | | 1 0 0 0 | 8 data bytes | |
| | | | 1 0 0 1 | 8 data bytes | 12 data bytes |
| | | | 1 0 1 0 | | 16 data bytes |
| | | | 1 0 1 1 | | 20 data bytes |
| | | | 1 1 0 0 | | 24 data bytes |
| | | | 1 1 0 1 | | 32 data bytes |
| 1 1 1 0 | | 48 data bytes | | | |
| 1 1 1 1 | | 64 data bytes | | | |
| 27 to 24 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. | | | |
| 23 to 16 | TMPTR[7:0] | Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer. | | | |
| 15 to 0 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. | | | |

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p ($p = m \times 16 + 15$) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCFDnCFDTMIDp register is set to 0 (data frame).

When the TMDLC[3:0] bits are set to 1001_B or more while the TMFDF bit in the RSCFDnCFDTMFDCTRp register is 0 (classical CAN frame), 8-byte data is transmitted actually. When the TMFDF bit is 1 (CAN FD frame), the settable value range varies depending on the setting of the TMME bit in the RSCFDnCFDCmFDCFG register.

- When the TMME bit = 0 (transmit buffer merge mode disabled):
A value of 0000_B to 1111_B can be set. If a value larger than 1100_B is set, payloads exceeding 20 bytes are padded by CCH.
- When the TMME bit = 1 (transmit buffer merge mode enabled):
When the corresponding transmit buffer number $p = (m \times 16) + 0$ or $(m \times 16) + 3$, a value of 0000_B to 1111_B can be set. In other cases, set a value of 0000_B to 1011_B (20 data bytes).

When the TMRTR bit is 1 (remote frame), these bits set the length of the message to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

17.4.11.5 RSCFDnCFDTMFDCTR_p — Transmit Buffer CAN FD Configuration Register _p (_p = 0 to 47)

Access: RSCFDnCFDTMFDCTR_p register can be read/written in 32-bit units
RSCFDnCFDTMFDCTR_{pL}, RSCFDnCFDTMFDCTR_{pH} registers can be read/written in 16-bit units
RSCFDnCFDTMFDCTR_{pLL}, RSCFDnCFDTMFDCTR_{pLH}, RSCFDnCFDTMFDCTR_{pHL},
RSCFDnCFDTMFDCTR_{pHH} registers can be read/written in 8-bit units

Address: RSCFDnCFDTMFDCTR_p: <RSCFDn_base> + 4008_H + (20_H × _p)
RSCFDnCFDTMFDCTR_{pL}: <RSCFDn_base> + 4008_H + (20_H × _p),
RSCFDnCFDTMFDCTR_{pH}: <RSCFDn_base> + 400A_H + (20_H × _p)
RSCFDnCFDTMFDCTR_{pLL}: <RSCFDn_base> + 4008_H + (20_H × _p),
RSCFDnCFDTMFDCTR_{pLH}: <RSCFDn_base> + 4009_H + (20_H × _p),
RSCFDnCFDTMFDCTR_{pHL}: <RSCFDn_base> + 400A_H + (20_H × _p),
RSCFDnCFDTMFDCTR_{pHH}: <RSCFDn_base> + 400B_H + (20_H × _p)

Initial value: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | TMFDF | TMBRS | TMESI |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 17.151 RSCFDnCFDTMFDCTR_p Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 2 | TMFDF | FD 0: Classical CAN frame 1: CAN FD frame |
| 1 | TMBRS | BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes. |
| 0 | TMESI | ESI 0: Error active node 1: Error passive node |

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (transmission not requested). When this register is linked to the transmit/receive FIFO buffer, do not write data to this register. When this register is allocated to the transmit queue, write data only to transmit buffer _p (_p = _m × 16 + 15) of the corresponding channel.

TMFDF Bit

This bit is used to set the FD format of the message to be transmitted from the transmit buffer.

TMBRS Bit

When this bit is set to 1 while the TMFDF bit is set to 1, the data area of a transmit message is transmitted at the data bit rate. When the TMFDF bit is set to 0, write 0 to this bit.

TMESI Bit

This bit is used to set the ESI bit value of the message to be transmitted from the transmit buffer when the TMFDF bit is set to 1. The set value is transmitted when the ESIC bit in the RSCFDnCFDCmFDCFG register is set to 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the TMFDF bit is set to 0, write 0 to this bit.

17.4.11.6 RSCFDnCFDTMDFb_p — Transmit Buffer Data Field b Register p (b = 0 to 4, p = 0 to 47)

Access: RSCFDnCFDTMDFb_p register can be read/written in 32-bit units
 RSCFDnCFDTMDFb_pL, RSCFDnCFDTMDFb_pH registers can be read/written in 16-bit units
 RSCFDnCFDTMDFb_pLL, RSCFDnCFDTMDFb_pLH, RSCFDnCFDTMDFb_pHL,
 RSCFDnCFDTMDFb_pHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTMDFb_p: <RSCFDn_base> + 400C_H + (04_H × b) + (20_H × p)
 RSCFDnCFDTMDFb_pL: <RSCFDn_base> + 400C_H + (04_H × b) + (20_H × p),
 RSCFDnCFDTMDFb_pH: <RSCFDn_base> + 400E_H + (04_H × b) + (20_H × p)
 RSCFDnCFDTMDFb_pLL: <RSCFDn_base> + 400C_H + (04_H × b) + (20_H × p),
 RSCFDnCFDTMDFb_pLH: <RSCFDn_base> + 400D_H + (04_H × b) + (20_H × p),
 RSCFDnCFDTMDFb_pHL: <RSCFDn_base> + 400E_H + (04_H × b) + (20_H × p),
 RSCFDnCFDTMDFb_pHH: <RSCFDn_base> + 400F_H + (04_H × b) + (20_H × p)

Initial value: 0000 0000_H

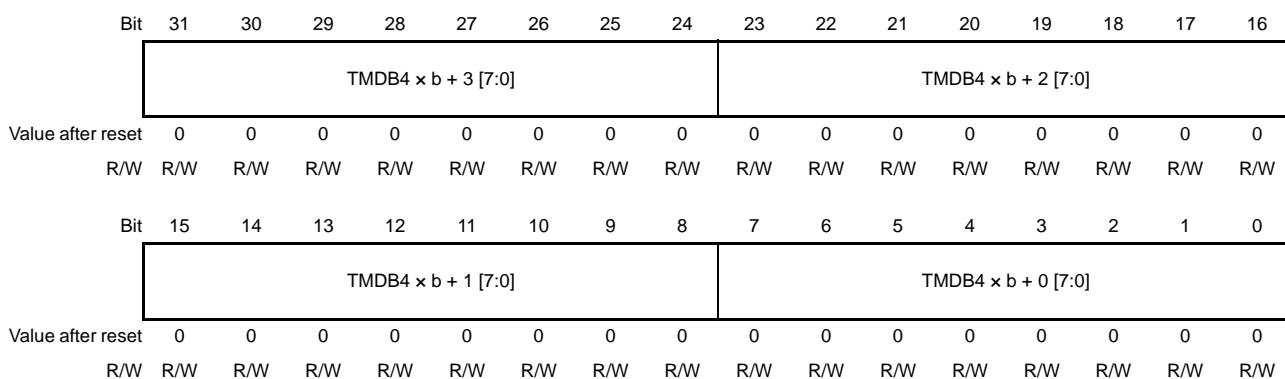


Table 17.152 RSCFDnCFDTMDFb_p Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|--|
| 31 to 24 | TMDB4 × b + 3 [7:0] | Transmit Buffer Data Byte 4 × b + 3 Transmit Buffer Data Byte 4 × b + 2 |
| 23 to 16 | TMDB4 × b + 2 [7:0] | Transmit Buffer Data Byte 4 × b + 1 Transmit Buffer Data Byte 4 × b + 0 |
| 15 to 8 | TMDB4 × b + 1 [7:0] | Set the transmit buffer data. |
| 7 to 0 | TMDB4 × b + 0 [7:0] | |

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

17.4.11.7 RSCFDnCFDTMIECy — Transmit Buffer Interrupt Enable Configuration Register y (y = 0, 1)

Access: RSCFDnCFDTMIECy register can be read/written in 32-bit units
 RSCFDnCFDTMIECyL, RSCFDnCFDTMIECyH registers can be read/written in 16-bit units
 RSCFDnCFDTMIECyLL, RSCFDnCFDTMIECyLH, RSCFDnCFDTMIECyHL, RSCFDnCFDTMIECyHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTMIECy: <RSCFDn_base> + 0390_H + (04_H × y)
 RSCFDnCFDTMIECyL: <RSCFDn_base> + 0390_H + (04_H × y),
 RSCFDnCFDTMIECyH: <RSCFDn_base> + 0392_H + (04_H × y)
 RSCFDnCFDTMIECyLL: <RSCFDn_base> + 0390_H + (04_H × y),
 RSCFDnCFDTMIECyLH: <RSCFDn_base> + 0391_H + (04_H × y),
 RSCFDnCFDTMIECyHL: <RSCFDn_base> + 0392_H + (04_H × y),
 RSCFDnCFDTMIECyHH: <RSCFDn_base> + 0393_H + (04_H × y)

Value after reset: 0000 0000_H

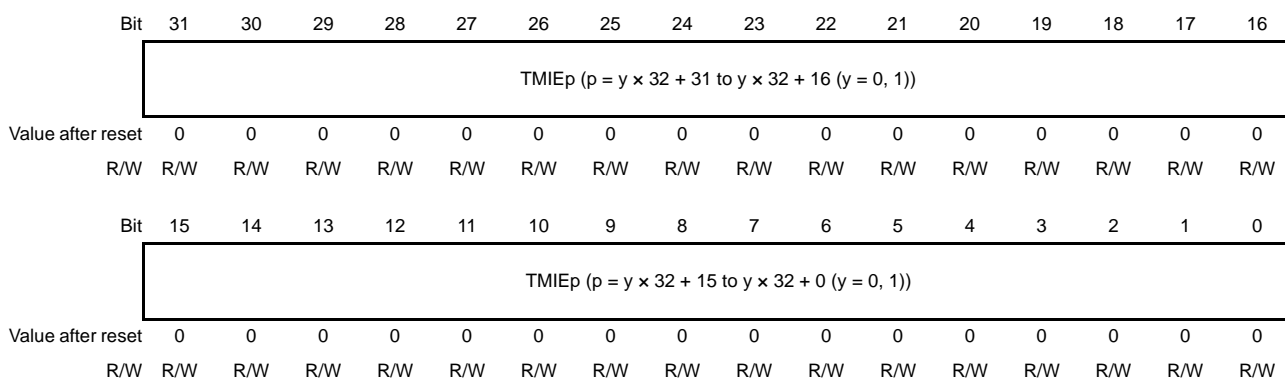


Table 17.153 RSCFDnCFDTMIECy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 16 | TMIEp | Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled |
| 15 to 0 | TMIEp | Transmit Buffer Interrupt Enable p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled. |

TMIEp Bits (p = 0 to 47)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCFDnCFDTMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

When the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode enable), set the bit corresponding to the transmit buffer allocated as a payload storage area to 0.

Table 17.154 shows the bit assignment.

Table 17.154 TMIEp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.4.12 Details of Transmit Buffer Status-related Registers

17.4.12.1 RSCFDnCFDTMTRSTSy — Transmit Buffer Transmit Request Status Register y (y = 0, 1)

Access: RSCFDnCFDTMTRSTSy register can be read only in 32-bit units
 RSCFDnCFDTMTRSTSyL, RSCFDnCFDTMTRSTSyH registers can be read only in 16-bit units
 RSCFDnCFDTMTRSTSyLL, RSCFDnCFDTMTRSTSyLH, RSCFDnCFDTMTRSTSyHL,
 RSCFDnCFDTMTRSTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTRSTSy: $\langle \text{RSCFDn_base} \rangle + 0350_{\text{H}} + (04_{\text{H}} \times y)$
 RSCFDnCFDTMTRSTSyL: $\langle \text{RSCFDn_base} \rangle + 0350_{\text{H}} + (04_{\text{H}} \times y)$,
 RSCFDnCFDTMTRSTSyH: $\langle \text{RSCFDn_base} \rangle + 0352_{\text{H}} + (04_{\text{H}} \times y)$
 RSCFDnCFDTMTRSTSyLL: $\langle \text{RSCFDn_base} \rangle + 0350_{\text{H}} + (04_{\text{H}} \times y)$,
 RSCFDnCFDTMTRSTSyLH: $\langle \text{RSCFDn_base} \rangle + 0351_{\text{H}} + (04_{\text{H}} \times y)$,
 RSCFDnCFDTMTRSTSyHL: $\langle \text{RSCFDn_base} \rangle + 0352_{\text{H}} + (04_{\text{H}} \times y)$,
 RSCFDnCFDTMTRSTSyHH: $\langle \text{RSCFDn_base} \rangle + 0353_{\text{H}} + (04_{\text{H}} \times y)$

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| TMTRSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMTRSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.155 RSCFDnCFDTMTRSTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | TMTRSTSp | Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit request is present. 1: A transmit request is present. |
| 15 to 0 | TMTRSTSp | Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present. |

TMTRSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTR bit in the RSCFDnCFDTMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 17.156 shows the bit assignment.

Table 17.156 TMTRSTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.4.12.2 RSCFDnCFDTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register y (y = 0, 1)

Access: RSCFDnCFDTMTARSTSy register can be read only in 32-bit units
 RSCFDnCFDTMTARSTSyL, RSCFDnCFDTMTARSTSyH registers can be read only in 16-bit units
 RSCFDnCFDTMTARSTSyLL, RSCFDnCFDTMTARSTSyLH, RSCFDnCFDTMTARSTSyHL,
 RSCFDnCFDTMTARSTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTARSTSy: <RSCFDn_base> + 0360_H + (04_H × y)
 RSCFDnCFDTMTARSTSyL: <RSCFDn_base> + 0360_H + (04_H × y),
 RSCFDnCFDTMTARSTSyH: <RSCFDn_base> + 0362_H + (04_H × y)
 RSCFDnCFDTMTARSTSyLL: <RSCFDn_base> + 0360_H + (04_H × y),
 RSCFDnCFDTMTARSTSyLH: <RSCFDn_base> + 0361_H + (04_H × y),
 RSCFDnCFDTMTARSTSyHL: <RSCFDn_base> + 0362_H + (04_H × y),
 RSCFDnCFDTMTARSTSyHH: <RSCFDn_base> + 0363_H + (04_H × y)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TMTARSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMTARSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.157 RSCFDnCFDTMTARSTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | TMTARSTSp | Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit abort request is present. 1: A transmit abort request is present. |
| 15 to 0 | TMTARSTSp | Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present. |

TMTARSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTAR bit in the RSCFDnCFDTMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 17.158 shows the bit assignment.

Table 17.158 TMTARSTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.4.12.3 RSCFDnCFDTMTCSTSy — Transmit Buffer Transmit Complete Status Register y (y = 0, 1)

Access: RSCFDnCFDTMTCSTSy register can be read only in 32-bit units
 RSCFDnCFDTMTCSTSyL, RSCFDnCFDTMTCSTSyH registers can be read only in 16-bit units
 RSCFDnCFDTMTCSTSyLL, RSCFDnCFDTMTCSTSyLH, RSCFDnCFDTMTCSTSyHL,
 RSCFDnCFDTMTCSTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTCSTSy: <RSCFDn_base> + 0370_H + (04_H × y)
 RSCFDnCFDTMTCSTSyL: <RSCFDn_base> + 0370_H + (04_H × y),
 RSCFDnCFDTMTCSTSyH: <RSCFDn_base> + 0372_H + (04_H × y)
 RSCFDnCFDTMTCSTSyLL: <RSCFDn_base> + 0370_H + (04_H × y),
 RSCFDnCFDTMTCSTSyLH: <RSCFDn_base> + 0371_H + (04_H × y),
 RSCFDnCFDTMTCSTSyHL: <RSCFDn_base> + 0372_H + (04_H × y),
 RSCFDnCFDTMTCSTSyHH: <RSCFDn_base> + 0373_H + (04_H × y)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| TMTTCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMTTCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.159 RSCFDnCFDTMTCSTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | TMTTCSTSp | Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed. |
| 15 to 0 | TMTTCSTSp | Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed. |

TMTTCSTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCFDnCFDTMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTTCSTSp flag is set to 1.

To clear the TMTTCSTSp flag to 0, set the corresponding TMTRF[1:0] flag to 00_B. This flag is cleared to 0 in channel reset mode.

Table 17.160 shows the bit assignment.

Table 17.160 TMCSTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.4.12.4 RSCFDnCFDTMTASTSy — Transmit Buffer Transmit Abort Status Register y (y = 0, 1)

Access: RSCFDnCFDTMTASTSy register can be read only in 32-bit units
 RSCFDnCFDTMTASTSyL, RSCFDnCFDTMTASTSyH registers can be read only in 16-bit units
 RSCFDnCFDTMTASTSyLL, RSCFDnCFDTMTASTSyLH, RSCFDnCFDTMTASTSyHL,
 RSCFDnCFDTMTASTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTASTSy: <RSCFDn_base> + 0380_H + (04_H × y)
 RSCFDnCFDTMTASTSyL: <RSCFDn_base> + 0380_H + (04_H × y),
 RSCFDnCFDTMTASTSyH: <RSCFDn_base> + 0382_H + (04_H × y)
 RSCFDnCFDTMTASTSyLL: <RSCFDn_base> + 0380_H + (04_H × y),
 RSCFDnCFDTMTASTSyLH: <RSCFDn_base> + 0381_H + (04_H × y),
 RSCFDnCFDTMTASTSyHL: <RSCFDn_base> + 0382_H + (04_H × y),
 RSCFDnCFDTMTASTSyHH: <RSCFDn_base> + 0383_H + (04_H × y)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TMTASTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1)) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.161 RSCFDnCFDTMTASTSy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | TMTASTSp | Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted |
| 15 to 0 | TMTASTSp | Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted. |

TMTASTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCFDnCFDTMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

To clear the TMTASTSp flag to 0, set the corresponding TMTRF[1:0] flag to 00_B. This flag is cleared to 0 in channel reset mode.

Table 17.162 shows the bit assignment.

Table 17.162 TMTASTSp Bit Assignment

| Bit Position | Channel | Transmit Buffer Number |
|--------------|---------|------------------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| . | . | . |
| . | . | . |
| 15 | 0 | 15 |
| 16 | 1 | 0 |
| . | . | . |
| . | . | . |
| 30 | 1 | 14 |
| 31 | 1 | 15 |
| 32 | 2 | 0 |
| 33 | 2 | 1 |
| . | . | . |
| . | . | . |
| 47 | 2 | 15 |

17.4.13 Details of Transmit Queue-related Registers

17.4.13.1 RSCFDnCFDTXQCCm — Transmit Queue Configuration and Control Register m (m = 0 to 2)

Access: RSCFDnCFDTXQCCm register can be read/written in 32-bit units
 RSCFDnCFDTXQCCmL, RSCFDnCFDTXQCCmH registers can be read/written in 16-bit units
 RSCFDnCFDTXQCCmLL, RSCFDnCFDTXQCCmLH, RSCFDnCFDTXQCCmHL, RSCFDnCFDTXQCCmHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTXQCCm: $\langle \text{RSCFDn_base} \rangle + 03A0_H + (04_H \times m)$
 RSCFDnCFDTXQCCmL: $\langle \text{RSCFDn_base} \rangle + 03A0_H + (04_H \times m)$,
 RSCFDnCFDTXQCCmH: $\langle \text{RSCFDn_base} \rangle + 03A2_H + (04_H \times m)$
 RSCFDnCFDTXQCCmLL: $\langle \text{RSCFDn_base} \rangle + 03A0_H + (04_H \times m)$,
 RSCFDnCFDTXQCCmLH: $\langle \text{RSCFDn_base} \rangle + 03A1_H + (04_H \times m)$,
 RSCFDnCFDTXQCCmHL: $\langle \text{RSCFDn_base} \rangle + 03A2_H + (04_H \times m)$,
 RSCFDnCFDTXQCCmHH: $\langle \text{RSCFDn_base} \rangle + 03A3_H + (04_H \times m)$

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|-------|-------|------------|-----|-----|-----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | TXQIM | TXQIE | TXQDC[3:0] | | | — | — | — | — | — | — | — | — | TXQE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R/W |

Table 17.163 RSCFDnCFDTXQCCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 14 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 13 | TXQIM | Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted. |
| 12 | TXQIE | Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled. |
| 11 to 8 | TXQDC[3:0] | Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g+1) transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited. For transmit buffer merge mode, set g to 2 to 9. |
| 7 to 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | TXQE | Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used. |

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from $(m \times 16 + 15)$ to $(m \times 16 + 0)$ (see **Table 17.93**). For examples of how buffer allocation is done, see **Figure 17.9**.

When the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode), transmit buffers $(m \times 16 + 5)$ to $(m \times 16 + 0)$ are merged and cannot be allocated to the transmit queue. Therefore, do not set TXQDC[3:0] bits to 10 to 15.

Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 enables the transmit queue. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010_B or more.

17.4.13.2 RSCFDnCFDTXQSTSm — Transmit Queue Status Register m (m = 0 to 2)

Access: RSCFDnCFDTXQSTSm register can be read/written in 32-bit units
 RSCFDnCFDTXQSTSmL, RSCFDnCFDTXQSTSmH registers can be read/written in 16-bit units
 RSCFDnCFDTXQSTSmLL, RSCFDnCFDTXQSTSmLH, RSCFDnCFDTXQSTSmHL,
 RSCFDnCFDTXQSTSmHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTXQSTSm: <RSCFDn_base> + 03C0_H + (04_H × m)
 RSCFDnCFDTXQSTSmL: <RSCFDn_base> + 03C0_H + (04_H × m),
 RSCFDnCFDTXQSTSmH: <RSCFDn_base> + 03C2_H + (04_H × m)
 RSCFDnCFDTXQSTSmLL: <RSCFDn_base> + 03C0_H + (04_H × m),
 RSCFDnCFDTXQSTSmLH: <RSCFDn_base> + 03C1_H + (04_H × m),
 RSCFDnCFDTXQSTSmHL: <RSCFDn_base> + 03C2_H + (04_H × m),
 RSCFDnCFDTXQSTSmHH: <RSCFDn_base> + 03C3_H + (04_H × m)

Value after reset: 0000 0001_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|------------|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQIF | TXQFL L | TXQEM P |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.164 RSCFDnCFDTXQSTSm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 13 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 12 to 8 | Reserved | When read, the undefined value is returned. When writing, write the value after reset. |
| 7 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | TXQIF | Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present. |
| 1 | TXQFL | Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full. |
| 0 | TXQEMP | Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty). |

TXQIF Flag

The TXQIF flag is set to 1 when the interrupt source specified by the TXQIM bit in the RSCFDnCFDTXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCFDnCFDTXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages stored in the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCFDnCFDTXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message stored in the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

17.4.13.3 RSCFDnCFDTXQPCTRM — Transmit Queue Pointer Control Register m (m = 0 to 2)

Access: RSCFDnCFDTXQPCTRM register can only be written in 32-bit units
 RSCFDnCFDTXQPCTRM_L, RSCFDnCFDTXQPCTRM_H registers can only be written in 16-bit units
 RSCFDnCFDTXQPCTRM_{LL}, RSCFDnCFDTXQPCTRM_{LH}, RSCFDnCFDTXQPCTRM_{HL},
 RSCFDnCFDTXQPCTRM_{HH} registers can only be written in 8-bit units

Address: RSCFDnCFDTXQPCTRM: <RSCFDn_base> + 03E0_H + (04_H × m)
 RSCFDnCFDTXQPCTRM_L: <RSCFDn_base> + 03E0_H + (04_H × m),
 RSCFDnCFDTXQPCTRM_H: <RSCFDn_base> + 03E2_H + (04_H × m)
 RSCFDnCFDTXQPCTRM_{LL}: <RSCFDn_base> + 03E0_H + (04_H × m),
 RSCFDnCFDTXQPCTRM_{LH}: <RSCFDn_base> + 03E1_H + (04_H × m),
 RSCFDnCFDTXQPCTRM_{HL}: <RSCFDn_base> + 03E2_H + (04_H × m),
 RSCFDnCFDTXQPCTRM_{HH}: <RSCFDn_base> + 03E3_H + (04_H × m)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | TXQPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.165 RSCFDnCFDTXQPCTRM Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 8 | Reserved | The write value should be the value after reset. |
| 7 to 0 | TXQPC[7:0] | Transmit Queue Pointer Control Writing FF _H to these bits moves the write pointer of the transmit queue to the next queue buffer. |

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request for the message. Write transmit messages to the RSCFDnCFDTMID_p, RSCFDnCFDTMPTR_p, RSCFDnCFDTMFDCTR_p, and RSCFDnCFDTMDFb_p registers (p = 15, 31, 47) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCFDnCFDTXQCC_m register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCFDnCFDTXQSTSM register is 0 (the transmit queue is not full).

17.4.14 Details of Transmit History-related Registers

17.4.14.1 RSCFDnCFDTHLCCm — Transmit History Configuration and Control Register m (m = 0 to 2)

Access: RSCFDnCFDTHLCCm register can be read/written in 32-bit units
 RSCFDnCFDTHLCCmL, RSCFDnCFDTHLCCmH registers can be read/written in 16-bit units
 RSCFDnCFDTHLCCmLL, RSCFDnCFDTHLCCmLH, RSCFDnCFDTHLCCmHL, RSCFDnCFDTHLCCmHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTHLCCm: $\langle \text{RSCFDn_base} \rangle + 0400_{\text{H}} + (04_{\text{H}} \times m)$
 RSCFDnCFDTHLCCmL: $\langle \text{RSCFDn_base} \rangle + 0400_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCFDnCFDTHLCCmH: $\langle \text{RSCFDn_base} \rangle + 0402_{\text{H}} + (04_{\text{H}} \times m)$
 RSCFDnCFDTHLCCmLL: $\langle \text{RSCFDn_base} \rangle + 0400_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCFDnCFDTHLCCmLH: $\langle \text{RSCFDn_base} \rangle + 0401_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCFDnCFDTHLCCmHL: $\langle \text{RSCFDn_base} \rangle + 0402_{\text{H}} + (04_{\text{H}} \times m)$,
 RSCFDnCFDTHLCCmHH: $\langle \text{RSCFDn_base} \rangle + 0403_{\text{H}} + (04_{\text{H}} \times m)$

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|------------|-------|-------|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | THLDT E | THLIM | THLIE | — | — | — | — | — | — | — | THLE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R | R | R/W |

Table 17.166 RSCFDnCFDTHLCCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 11 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 10 | THLDTE | Transmit History Target Buffer Select 0: Entries from transmit/receive FIFO buffers and transmit queue 1: Entries from transmit buffers, transmit/receive FIFO buffers, and transmit queue |
| 9 | THLIM | Transmit History Interrupt Source Select 0: When 12 items of data have been stored in the transmit history buffer 1: Each time transmit history data is stored in the transmit history buffer |
| 8 | THLIE | Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled. |
| 7 to 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | THLE | Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used. |

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit is set to 0.

THLE Bit

Setting this bit to 1 enables the transmit history buffer. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in channel reset mode.

17.4.14.2 RSCFDnCFDTHLSTSm — Transmit History Status Register m (m = 0 to 2)

Access: RSCFDnCFDTHLSTSm register can be read/written in 32-bit units
 RSCFDnCFDTHLSTSmL, RSCFDnCFDTHLSTSmH register can be read/written in 16-bit units
 RSCFDnCFDTHLSTSmLL, RSCFDnCFDTHLSTSmLH, RSCFDnCFDTHLSTSmHL,
 RSCFDnCFDTHLSTSmHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTHLSTSm: <RSCFDn_base> + 0420_H + (04_H × m)
 RSCFDnCFDTHLSTSmL: <RSCFDn_base> + 0420_H + (04_H × m),
 RSCFDnCFDTHLSTSmH: <RSCFDn_base> + 0422_H + (04_H × m)
 RSCFDnCFDTHLSTSmLL: <RSCFDn_base> + 0420_H + (04_H × m),
 RSCFDnCFDTHLSTSmLH: <RSCFDn_base> + 0421_H + (04_H × m),
 RSCFDnCFDTHLSTSmHL: <RSCFDn_base> + 0422_H + (04_H × m),
 RSCFDnCFDTHLSTSmHH: <RSCFDn_base> + 0423_H + (04_H × m)

Value after reset: 0000 0001_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|------------|----|----|----|----|----|----|----|-------|--------|--------|--------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | THLMC[4:0] | | | | — | — | — | — | THLIF | THLELT | THLFLL | THLEMP | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W*1 | R/W*1 | R | R |

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 17.167 RSCFDnCFDTHLSTSm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 13 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 12 to 8 | THLMC[4:0] | Transmit History Buffer Unread Data Counter These bits indicate the number of unread data stored in the transmit history buffer. |
| 7 to 4 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 3 | THLIF | Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present. |
| 2 | THLELT | Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred. |
| 1 | THLFLL | Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full. |
| 0 | THLEMP | Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty). |

THLMC[4:0] Bits

These bits indicate the number of unread data stored in the transmit history buffer. These bits are cleared to 0 in channel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCFDnCFDTHLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 16 items of data have been stored in the transmit history buffer, and is cleared to 0 when the number of data stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCFDnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single item of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCFDnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

17.4.14.3 RSCFDnCFDTHLPCTRm — Transmit History Pointer Control Register m (m = 0 to 2)

Access: RSCFDnCFDTHLPCTRm register can only be written in 32-bit units
RSCFDnCFDTHLPCTRmL, RSCFDnCFDTHLPCTRmH registers can only be written in 16-bit units
RSCFDnCFDTHLPCTRmLL, RSCFDnCFDTHLPCTRmLH, RSCFDnCFDTHLPCTRmHL,
RSCFDnCFDTHLPCTRmHH registers can only be written in 8-bit units

Address: RSCFDnCFDTHLPCTRm: <RSCFDn_base> + 0440_H + (04_H × m)
RSCFDnCFDTHLPCTRmL: <RSCFDn_base> + 0440_H + (04_H × m),
RSCFDnCFDTHLPCTRmH: <RSCFDn_base> + 0442_H + (04_H × m)
RSCFDnCFDTHLPCTRmLL: <RSCFDn_base> + 0440_H + (04_H × m),
RSCFDnCFDTHLPCTRmLH: <RSCFDn_base> + 0441_H + (04_H × m),
RSCFDnCFDTHLPCTRmHL: <RSCFDn_base> + 0442_H + (04_H × m),
RSCFDnCFDTHLPCTRmHH: <RSCFDn_base> + 0443_H + (04_H × m)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | THLPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 17.168 RSCFDnCFDTHLPCTRm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 8 | Reserved | When writing, write the value after reset. |
| 7 to 0 | THLPC[7:0] | Transmit History List Pointer Control Writing FF _H to these bits moves the read pointer to the next unread data in the transmit history buffer. |

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCFDnCFDTHLSTSm register is decremented by 1. Write FF_H to the THLPC[7:0] bits after reading from the RSCFDnCFDTHLACCm register.

When writing FF_H to these bits, make sure that the THLE bit in the RSCFDnCFDTHLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCFDnCFDTHLSTSm register is 0.

17.4.14.4 RSCFDnCFDTHLACCm — Transmit History Access Register m (m = 0 to 2)

Access: RSCFDnCFDTHLACCm register can be read only in 32-bit units
 RSCFDnCFDTHLACCmL, RSCFDnCFDTHLACCmH registers can be read only in 16-bit units
 RSCFDnCFDTHLACCmLL, RSCFDnCFDTHLACCmLH, RSCFDnCFDTHLACCmHL,
 RSCFDnCFDTHLACCmHH registers can be read only in 8-bit units

Address: RSCFDnCFDTHLACCm: <RSCFDn_base> + 6000_H + (04_H × m)
 RSCFDnCFDTHLACCmL: <RSCFDn_base> + 6000_H + (04_H × m),
 RSCFDnCFDTHLACCmH: <RSCFDn_base> + 6002_H + (04_H × m)
 RSCFDnCFDTHLACCmLL: <RSCFDn_base> + 6000_H + (04_H × m),
 RSCFDnCFDTHLACCmLH: <RSCFDn_base> + 6001_H + (04_H × m),
 RSCFDnCFDTHLACCmHL: <RSCFDn_base> + 6002_H + (04_H × m),
 RSCFDnCFDTHLACCmHH: <RSCFDn_base> + 6003_H + (04_H × m)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|---------|----|----|---------|----|----|----|----|
| | TMTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TID[7:0] | | | | | | | — | BN[3:0] | | | BT[2:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.169 RSCFDnCFDTHLACCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 16 | TMTS[15:0] | Timestamp Data The timestamp data of stored data can be read. |
| 15 to 8 | TID[7:0] | Label Data The label information of stored data can be read. |
| 7 | Reserved | When read, the value after reset is returned. |
| 6 to 3 | BN[3:0] | Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read. |
| 2 to 0 | BT[2:0] | Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue |

TMTS[15:0] Bits

Timestamp values in transmit history data stored in the transmit history buffer are displayed.

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

17.4.15 Details of Test-related Registers

17.4.15.1 RSCFDnCFDGTSTCFG — Global Test Configuration Register

Access: RSCFDnCFDGTSTCFG register can be read/written in 32-bit units
 RSCFDnCFDGTSTCFG_L, RSCFDnCFDGTSTCFG_H registers can be read/written in 16-bit units
 RSCFDnCFDGTSTCFG_{LL}, RSCFDnCFDGTSTCFG_{LH}, RSCFDnCFDGTSTCFG_{HL},
 RSCFDnCFDGTSTCFG_{HH} registers can be read/written in 8-bit units

Address: RSCFDnCFDGTSTCFG: <RSCFDn_base> + 0468_H
 RSCFDnCFDGTSTCFG_L: <RSCFDn_base> + 0468_H, RSCFDnCFDGTSTCFG_H: <RSCFDn_base> + 046A_H
 RSCFDnCFDGTSTCFG_{LL}: <RSCFDn_base> + 0468_H, RSCFDnCFDGTSTCFG_{LH}: <RSCFDn_base> +
 0469_H,
 RSCFDnCFDGTSTCFG_{HL}: <RSCFDn_base> + 046A_H, RSCFDnCFDGTSTCFG_{HH}: <RSCFDn_base> +
 046B_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|------------|-----|-----|-----|---------|---------|---------|
| | — | — | — | — | — | — | — | — | — | RTMPS[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | C2ICBCE | C1ICBCE | C0ICBCE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 17.170 RSCFDnCFDGTSTCFG Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 23 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 22 to 16 | RTMPS[6:0] | RAM Test Page Configuration Set a value within a range of page 0 (00 _H) to page 41 (29 _H). |
| 15 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | C2ICBCE | CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled. 1: CAN2 inter-channel communication test is enabled. |
| 1 | C1ICBCE | CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled. |
| 0 | C0ICBCE | CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled. |

Modify the RSCFDnCFDGTSTCFG register only in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to 29_H, inclusive. In CAN FD mode, do not access more than 160 bytes in the last page (RTMPS = 29_H) during RAM test.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

17.4.15.2 RSCFDnCFDGTSTCTR — Global Test Control Register

Access: RSCFDnCFDGTSTCTR register can be read/written in 32-bit units
 RSCFDnCFDGTSTCTRL, RSCFDnCFDGTSTCTRH registers can be read/written in 16-bit units
 RSCFDnCFDGTSTCTRL, RSCFDnCFDGTSTCTRLH, RSCFDnCFDGTSTCTRHL,
 RSCFDnCFDGTSTCTRHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGTSTCTR: <RSCFDn_base> + 046C_H
 RSCFDnCFDGTSTCTRL: <RSCFDn_base> + 046C_H, RSCFDnCFDGTSTCTRH: <RSCFDn_base> + 046E_H
 RSCFDnCFDGTSTCTRL: <RSCFDn_base> + 046D_H, RSCFDnCFDGTSTCTRLH: <RSCFDn_base> +
 046D_H,
 RSCFDnCFDGTSTCTRHL: <RSCFDn_base> + 046E_H, RSCFDnCFDGTSTCTRHH: <RSCFDn_base> +
 046F_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|---------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RTME | — | ICBCTME |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R | R/W |

Table 17.171 RSCFDnCFDGTSTCTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 2 | RTME | RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled. |
| 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | ICBCTME | Inter-Channel Communication Test Enable 0: Inter-channel communication test is disabled. 1: Inter-channel communication test is enabled. |

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode (see **Figure 17.37**).

1. Set the GMDC[1:0] bits in the RSCFDnCFDGTCTR register to 10_B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 2) in the RSCFDnCFDGTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

This bit is cleared to 0 in global reset mode.

17.4.15.3 RSCFDnCFDGLOCKK — Global Lock Key Register

Access: RSCFDnCFDGLOCKK register can be write only in 32-bit units.
RSCFDnCFDGLOCKKL, RSCFDnCFDGLOCKKH registers can be write only in 16-bit units.

Address: RSCFDnCFDGLOCKK: <RSCFDn_base> + 047C_H
RSCFDnCFDGLOCKKL: <RSCFDn_base> + 047C_H, RSCFDnCFDGLOCKKH: <RSCFDn_base> + 047E_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LOCK[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 | W*1 |

Note 1. Writing to these bits is effective only when the RS-CANFD module is in global test mode.

Table 17.172 RSCFDnCFDGLOCKK Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 16 | Reserved | When writing these bits, write the value after reset. |
| 15 to 0 | LOCK[15:0] | Lock Key These bits are key bits to release protection of test mode. |

The RSCFDnCFDGLOCKK register releases protection of special test bits and is write only.

For the protection release data, see **Section 17.11.4.2, Procedure for Releasing the Protection.**

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCFDnCFDGTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCFDn_base> + 0000_H to <RSCFDn_base> + 05FF_H) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

17.4.15.4 RSCFDnCFDRPGACC_r — RAM Test Page Access Register *r* (*r* = 0 to 63)

Access: RSCFDnCFDRPGACC_r register can be read/written in 32-bit units
 RSCFDnCFDRPGACC_{rL}, RSCFDnCFDRPGACC_{rH} registers can be read/written in 16-bit units
 RSCFDnCFDRPGACC_{rLL}, RSCFDnCFDRPGACC_{rLH}, RSCFDnCFDRPGACC_{rHL},
 RSCFDnCFDRPGACC_{rHH} registers can be read/written in 8-bit units

Address: RSCFDnCFDRPGACC_r: <RSCFDn_base> + 6400_H + (04_H × *r*)
 RSCFDnCFDRPGACC_{rL}: <RSCFDn_base> + 6400_H + (04_H × *r*),
 RSCFDnCFDRPGACC_{rH}: <RSCFDn_base> + 6402_H + (04_H × *r*)
 RSCFDnCFDRPGACC_{rLL}: <RSCFDn_base> + 6400_H + (04_H × *r*),
 RSCFDnCFDRPGACC_{rLH}: <RSCFDn_base> + 6401_H + (04_H × *r*),
 RSCFDnCFDRPGACC_{rHL}: <RSCFDn_base> + 6402_H + (04_H × *r*),
 RSCFDnCFDRPGACC_{rHH}: <RSCFDn_base> + 6403_H + (04_H × *r*)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RDTA[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RDTA[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 17.173 RSCFDnCFDRPGACC_r Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 0 | RDTA[31:0] | RAM Data Test Access RAM data for CAN can be read and written. |

Modify the RSCFDnCFDRPGACC_r register in global test mode with the RTME bit in the RSCFDnCFDGTSTCTR register set to 1 (RAM test is enabled).

The RSCFDnCFDRPGACC_r register can be read and written when the RTME bit is set to 1.

17.4.16 Details of Mode Read Register

17.4.16.1 RSCFDnCANFDMDR — CAN FD Mode Read Register

Access: This register can be read in 32-bit units

Address: <RSCFDn_base> + 8000_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FDMDR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 17.174 RSCFDnCANFDMDR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | These bits are read as the value after reset. The write value should be the value after reset. |
| 0 | FDMDR | CAN FD mode read 0: classical CAN mode is selected. 1: CAN FD mode is selected. |

17.5 Interrupt Sources and DMA Trigger

17.5.1 Interrupt Sources

The RS-CANFD module has 11 interrupts that are grouped into global interrupts and channel interrupts.

- Global interrupts (2 sources):
 - Receive FIFO interrupt
 - Global error interrupt
- Channel interrupts (3 sources/channel):
 - CAN_m transmit interrupt (m = 0 to 2)
 - CAN_m transmit complete interrupt
 - CAN_m transmit abort interrupt
 - CAN_m transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)
 - CAN_m transmit history interrupt
 - CAN_m transmit queue Interrupt
 - CAN_m transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode)
 - CAN_m error interrupt

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In this case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the RS-CANFD module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The current interrupt request is still output until the interrupt request flag is cleared.

Table 17.175 lists the CAN interrupt sources. **Figure 17.2** shows the CAN global interrupt block diagram. **Figure 17.3** shows the CAN channel interrupt block diagram.

Table 17.175 List of CAN Interrupt Sources

| | Interrupt Source | | Corresponding Interrupt Request Flag | Corresponding Interrupt Enable Bit |
|------------------------------------|------------------|--|---|---|
| Global interrupts | Receive FIFO | Receive FIFO 0 | RFIF in the RSCFDn(CFD)RFSTS0 register | RFIE in the RSCFDn(CFD)RFCC0 register |
| | | Receive FIFO 1 | RFIF in the RSCFDn(CFD)RFSTS1 register | RFIE in the RSCFDn(CFD)RFCC1 register |
| | | Receive FIFO 2 | RFIF in the RSCFDn(CFD)RFSTS2 register | RFIE in the RSCFDn(CFD)RFCC2 register |
| | | Receive FIFO 3 | RFIF in the RSCFDn(CFD)RFSTS3 register | RFIE in the RSCFDn(CFD)RFCC3 register |
| | | Receive FIFO 4 | RFIF in the RSCFDn(CFD)RFSTS4 register | RFIE in the RSCFDn(CFD)RFCC4 register |
| | | Receive FIFO 5 | RFIF in the RSCFDn(CFD)RFSTS5 register | RFIE in the RSCFDn(CFD)RFCC5 register |
| | | Receive FIFO 6 | RFIF in the RSCFDn(CFD)RFSTS6 register | RFIE in the RSCFDn(CFD)RFCC6 register |
| | | Receive FIFO 7 | RFIF in the RSCFDn(CFD)RFSTS7 register | RFIE in the RSCFDn(CFD)RFCC7 register |
| | Global error | <ul style="list-style-type: none"> DEF in the RSCFDn(CFD)GERFL register MES in the RSCFDn(CFD)GERFL register THLES in the RSCFDn(CFD)GERFL register CMPOF in the RSCFDn(CFD)GERFL register | <ul style="list-style-type: none"> DEIE in the RSCFDn(CFD)GCTR register MEIE in the RSCFDn(CFD)GCTR register THLEIE in the RSCFDn(CFD)GCTR register CMPOFIE in the RSCFDn(CFD)GCTR register | |
| Channel interrupts (m = 0 to 2) | CANm transmit | CANm transmit complete | TMTRF[1:0] in the RSCFDn(CFD)TMSTSp register | TMIEp in the RSCFDn(CFD)TMIECy register |
| | | CANm transmit abort | TMTRF[1:0] in the RSCFDn(CFD)TMSTSp register | TAIE in the RSCFDn(CFD)CmCTR register |
| | | CANm transmit/receive FIFO transmit complete | CFTXIF in the RSCFDn(CFD)CFSTSk register | CFTXIE in the RSCFDn(CFD)CFCCk register |
| | | CANm transmit queue | TXQIF in the RSCFDn(CFD)TXQSTSm register | TXQIE in the RSCFDn(CFD)TXQCCm register |
| | | CANm transmit history | THLIF in the RSCFDn(CFD)THLSTSm register | THLIE in the RSCFDn(CFD)THLCCm register |
| | | CANm transmit/receive FIFO receive complete | CFRXIF in the RSCFDn(CFD)CFSTSk register | CFRXIE in the RSCFDn(CFD)CFCCk register |
| | CANm error | <ul style="list-style-type: none"> BEF in the RSCFDn(CFD)CmERFL register ALF in the RSCFDn(CFD)CmERFL register BLF in the RSCFDn(CFD)CmERFL register OVLf in the RSCFDn(CFD)CmERFL register BORF in the RSCFDn(CFD)CmERFL register BOEF in the RSCFDn(CFD)CmERFL register EPF in the RSCFDn(CFD)CmERFL register EWf in the RSCFDn(CFD)CmERFL register SOCO in the RSCFDn(CFD)CFDCmFDSTS register EOCO in the RSCFDn(CFD)CFDCmFDSTS register TDCVF in the RSCFDn(CFD)CFDCmFDSTS register | <ul style="list-style-type: none"> BEIE in the RSCFDn(CFD)CmCTR register ALIE in the RSCFDn(CFD)CmCTR register BLIE in the RSCFDn(CFD)CmCTR register OLIE in the RSCFDn(CFD)CmCTR register BORIE in the RSCFDn(CFD)CmCTR register BOEIE in the RSCFDn(CFD)CmCTR register EPIE in the RSCFDn(CFD)CmCTR register EWIE in the RSCFDn(CFD)CmCTR register SOCOIE in the RSCFDn(CFD)CFDCmCTR register EOCOIE in the RSCFDn(CFD)CFDCmCTR register TDCVFIE in the RSCFDn(CFD)CFDCmCTR register | |

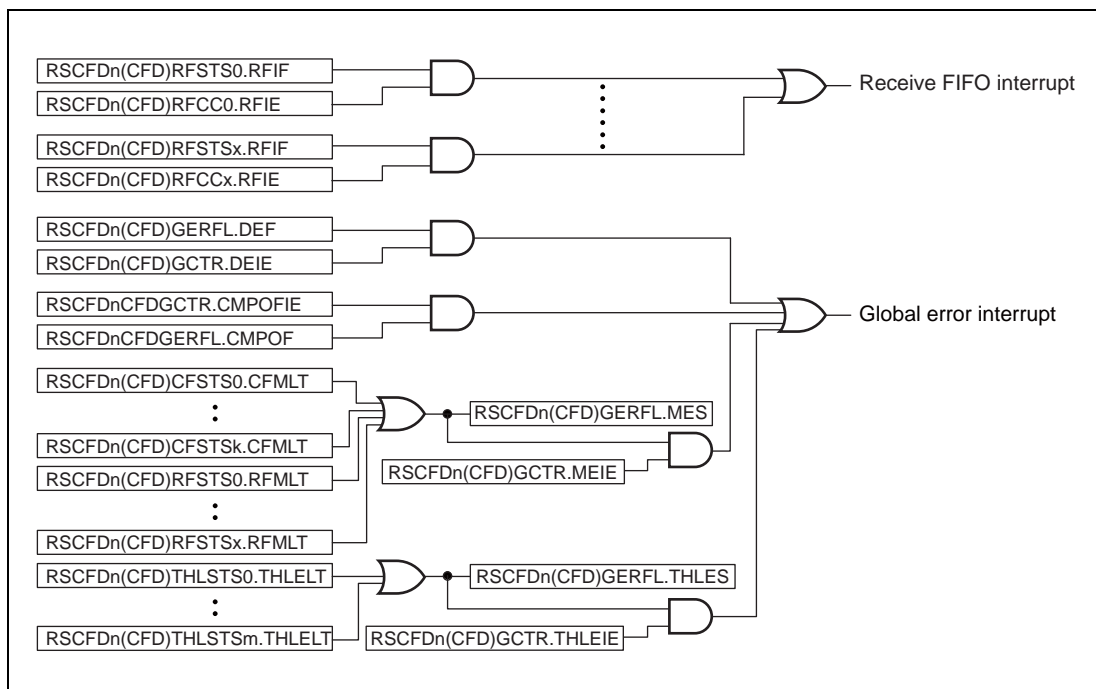


Figure 17.2 CAN Global Interrupt Block Diagram

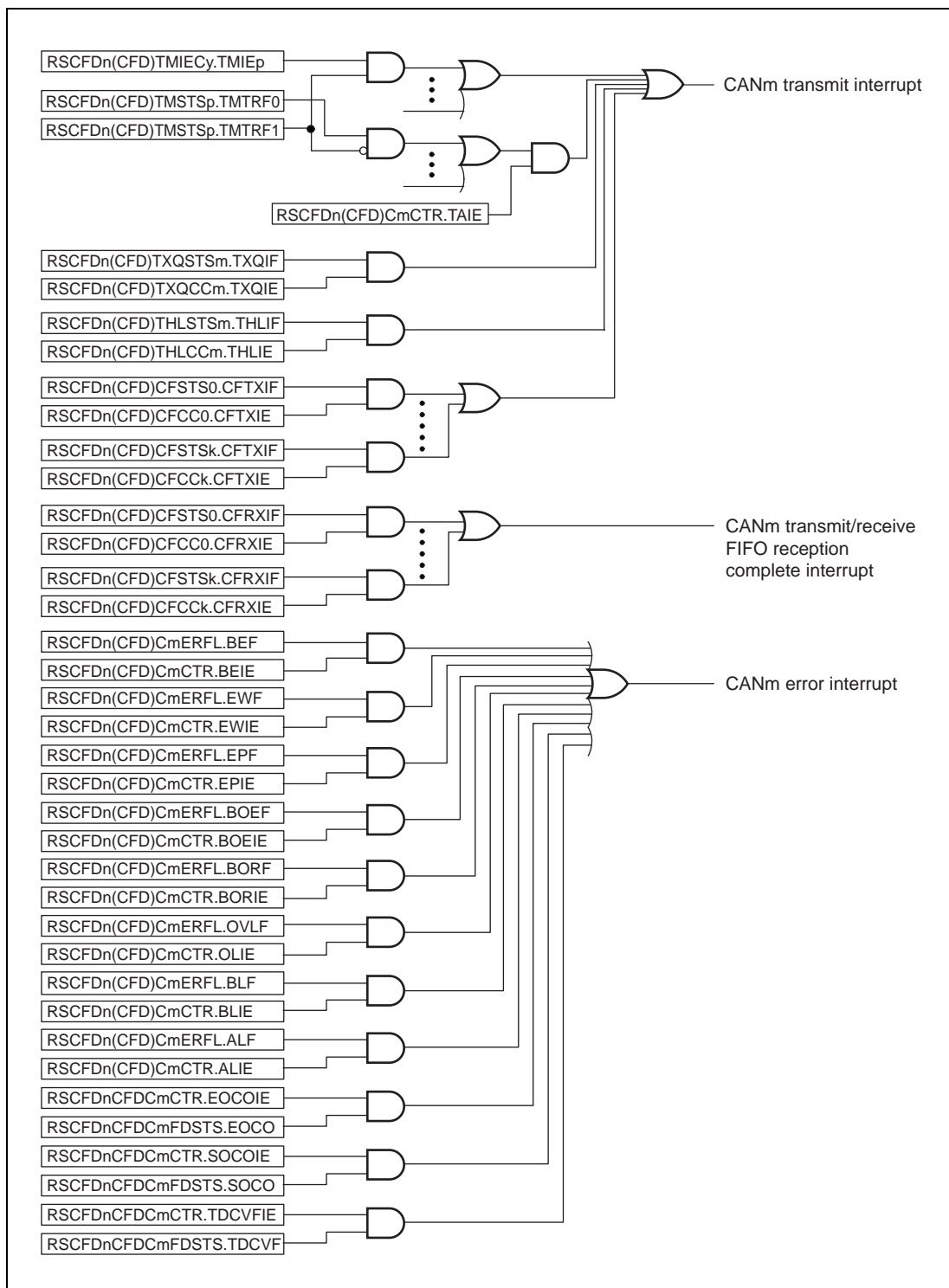


Figure 17.3 CAN Channel Interrupt Block Diagram

17.5.2 DMA Trigger (Only in CAN FD Mode)

In CAN FD mode, receive FIFO buffers can be related to DMA channels. The following 14 FIFO buffers can be related.

- All receive FIFO buffers x ($x = 0$ to 7)
- The first transmit/receive FIFO buffer k ($k = 3 \times m$, $m = 0$ to 2) allocated to channel m

When the DMA enable bit (RFDMAEx or CFDMAEm bit in the RSCFDnCFDCDTCT register) is set to 1 and an unread message is remaining in the related FIFO, a DMA transfer request trigger is generated.

17.6 CAN Modes

The RS-CANFD module has four global modes to control the entire RS-CANFD module status and four channel modes to control individual channel status. Details of global modes are described in **Section 17.6.1, Global Modes**, and details of channel modes are described in **Section 17.6.2, Channel Modes**.

- Global stop mode: Stops the clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs the RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

17.6.1 Global Modes

Figure 17.4 shows the transitions of global modes.

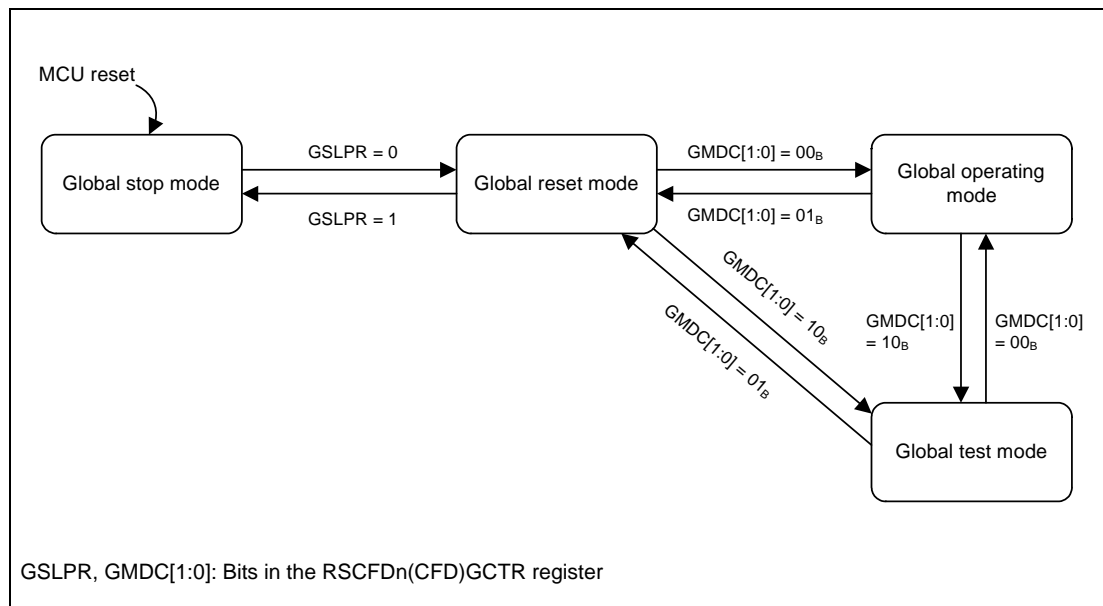


Figure 17.4 Transitions of Global Modes

In some cases, global mode transitions also force channel mode transitions. **Table 17.176** shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

Table 17.176 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)

| Channel Mode before Setting | Channel Mode after Setting | | | |
|-----------------------------|--|---|--|---|
| | GMDC[1:0] = 00 _B GSLPR = 0 (Global Operation) | GMDC[1:0] = 10 _B GSLPR = 0 (Global Test) | GMDC[1:0] = 01 _B GSLPR = 0 (Global Reset) | GMDC[1:0] = 01 _B GSLPR = 1 (Global Stop) |
| Channel communication | Channel communication | Channel halt | Channel reset | Transition prohibited |
| Channel halt | Channel halt | Channel halt | Channel reset | Transition prohibited |
| Channel reset | Channel reset | Channel reset | Channel reset | Channel stop |
| Channel stop | Channel stop | Channel stop | Channel stop | Channel stop |

Note: GMDC[1:0], GSLPR: Bits in the RSCFDn(CFD)GCTR register

Table 17.177 shows the global mode transition time.

Table 17.177 Global Mode Transition Time

| Mode before Transition | Mode after Transition | Maximum Transition Time |
|------------------------|-----------------------|------------------------------------|
| Global stop | Global reset | Three pclk cycles |
| Global reset | Global stop | Three pclk cycles |
| Global reset | Global test | Ten pclk cycles |
| Global reset | Global operating | Ten pclk cycles |
| Global test | Global reset | Two CAN bit times ^{*1,*2} |
| Global test | Global operating | Three pclk cycles |
| Global operating | Global reset | Two CAN bit times ^{*1,*2} |
| Global operating | Global test | Two CAN frames ^{*1} |

Note 1. CAN frame time and CAN bit time of the lowest communication speed of the channels in use

Note 2. In CAN FD mode, this time value is the CAN bit time of the nominal bit rate.

17.6.1.1 Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. Only the clock used by the CPU for writing to the GSLPR bit runs in this mode.

After the MCU is reset, the CAN module transitions to global stop mode. Setting the GSLPR bit in the RSCFDn(CFD)GCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each RSCFDn(CFD)CmCTR register to 1 (channel stop mode). Afterwards, if all channels are forced to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

17.6.1.2 Global Reset Mode

In global reset mode, RS-CANFD module settings are performed. When the RS-CANFD module transitions to global reset mode, some registers are initialized. For registers to be initialized, see **Table 17.180, Registers Initialized in Global Reset Mode or Channel Reset Mode** and **Table 17.181, Registers Initialized Only in Global Reset Mode**.

Setting the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register to 01_B sets the CHMDC[1:0] bits in each RSCFDn(CFD)CmCTR register (m = 0 to 2) to 01_B (channel reset mode). If all channels are forced to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to 01_B).

17.6.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register to 10_B sets the CHMDC[1:0] bits in each RSCFDn(CFD)CmCTR register to 10_B (channel halt mode). If all channels are forced to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

17.6.1.4 Global Operating Mode

The RS-CANFD module operates in global operating mode.

When the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register are set to 00_B, the RS-CANFD module transitions to global operating mode.

17.6.2 Channel Modes

Figure 17.5 shows a channel mode state transition chart. Table 17.178 shows the channel mode transition time.

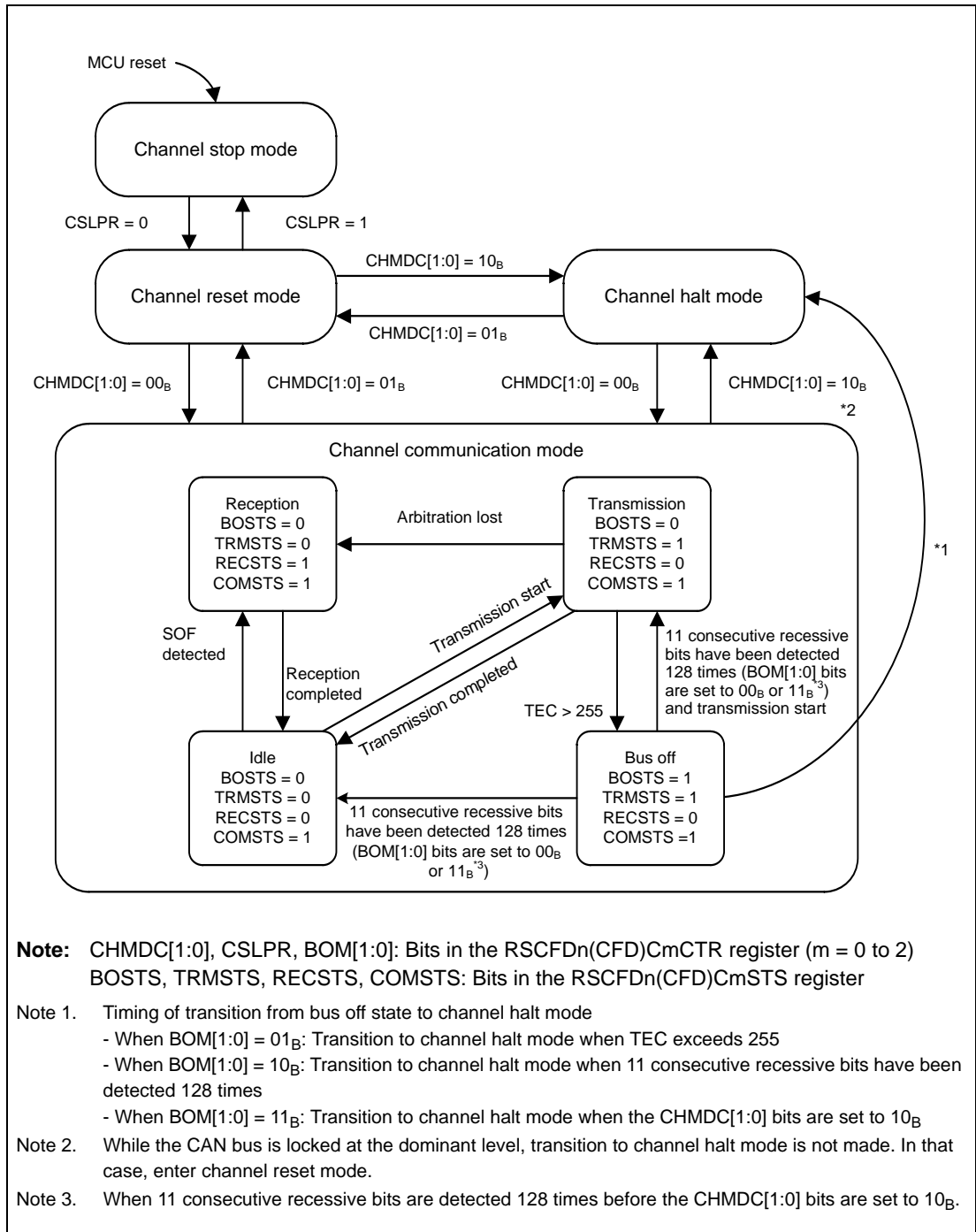


Figure 17.5 Channel Mode State Transition Chart

Table 17.178 Channel Mode Transition Time

| Mode before Transition | Mode after Transition | Maximum Transition Time |
|------------------------|-----------------------|------------------------------------|
| Channel stop | Channel reset | Three pclk cycles |
| Channel reset | Channel stop | Three pclk cycles |
| Channel reset | Channel halt | Three CANm bit times* ¹ |
| Channel reset | Channel communication | Four CANm bit times* ¹ |
| Channel halt | Channel reset | Two CANm bit times* ¹ |
| Channel halt | Channel communication | Four CANm bit times* ¹ |
| Channel communication | Channel reset | Two CANm bit times* ¹ |
| Channel communication | Channel halt | Two CANm frames |

Note 1. In CAN FD mode, this time value is the CANm bit time of the nominal bit rate.

17.6.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. Channel-related registers can be read, but writing data to them is prohibited (except write to CSLPR bit). Register values are retained.

Each channel enters channel stop mode after the MCU is reset. Channels also transition to channel stop mode when the CSLPR bit in the RSCFDn(CFD)CmCTR register (m = 0 to 2) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

17.6.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. For registers to be initialized, see **Table 17.180, Registers Initialized in Global Reset Mode or Channel Reset Mode.**

When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 01_B (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. **Table 17.179** shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

17.6.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 17.179 shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 17.179 Operation when a Channel Transitions to Channel Reset Mode/Channel Halt Mode

| Mode | During Reception | During Transmission | Bus Off State |
|---|--|---|--|
| Channel reset (CHMDC[1:0] = 01 _B) | Transitions to channel reset mode before reception is completed.*1 | Transitions to channel reset mode before transmission is completed.*1 | Transitions to channel reset mode before bus off recovery. |
| Channel halt*3 (CHMDC[1:0] = 10 _B) | Transitions to channel halt mode after reception is completed.*2 | Transitions to channel halt mode after transmission is completed. | [When BOM[1:0] = 00 _B] Transitions to channel halt mode (CHMDC[1:0] = 10 _B) only after bus off recovery. [When BOM[1:0] = 01 _B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10 _B] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] = 11 _B] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10 _B before bus off recovery. |

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10_B and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01_B.

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RSCFDn(CFD)CmERFL register that becomes 1 when bus lock is detected.

Note 3. In classical CAN mode, when the transition from channel reset mode to channel halt mode is to be made, set the RSCANnCmCFG register in channel reset mode and then transition to channel halt mode. In CAN FD mode, set the RSCFDnCFDCmNCFG register and the RSCFDnCFDCmDCFG register, and then make a transition.

17.6.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception: Receiving a message sent from another node.
- Transmission: Transmitting a message.
- Bus off: Isolated from CAN communication.

When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 00_B, the channel transitions to channel communication mode. After that, once 11 consecutive recessive bits have been detected, the COMSTS flag in the RSCFDn(CFD)CmSTS register (m = 0 to 2) is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

17.6.2.5 Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications.

The conditions for returning from the bus off state are determined by the BOM[1:0] bits in the RSCFDn(CFD)CmCTR register.

- When BOM[1:0] = 00_B:
Bus off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RSCFDn(CFD)CmSTS register are initialized to 00_H, the BORF flag in the RSCFDn(CFD)CmERFL register is set to 1 (bus off recovery is detected), and a bus off recovery interrupt request is generated. When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 10_B (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When BOM[1:0] = 01_B:
When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to 10_B and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H. The BORF flag is not set to 1, and bus off recovery interrupt request is not generated.
- When BOM[1:0] = 10_B:
When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to 10_B. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, the BORF flag is set to 1, and a bus off recovery interrupt request is generated.
- When BOM[1:0] = 11_B:
When the CHMDC[1:0] bits are set to 10_B in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, but the BORF flag is not set to 1. Also, a bus off recovery interrupt is not generated.
However, the BORF flag becomes 1 and a bus off recovery interrupt request is generated if a CAN module transitions to error active state (by detecting 11 consecutive recessive bits 128 times) before CHMDC[1:0] bits are set to 10_B.

If the RS-CANFD module causes the channel to transition to channel halt mode simultaneously with a program write to the CHMDC[1:0] bits, the program write takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to 01_B or 10_B is made only when the CHMDC[1:0] bits are 00_B (channel communication mode).

Furthermore, setting the RTBO bit in the RSCFDn(CFD)CmCTR register to 1 allows a forced return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to 00_H. Write 1 to the RTBO bit only when the BOM[1:0] value is 00_B. Writing 1 to the RTBO bit in a state other than the bus off state is ignored, and the RTBO bit is immediately set to 0.

17.6.3 Initializing Registers by Transition to CAN Mode

Table 17.180 lists bits and flags to be initialized by a transition to channel reset mode. These bits and flags are also initialized by a transition to global reset mode. Furthermore, **Table 17.181** lists bits and flags to be initialized only by a transition to global reset mode.

Table 17.180 Registers Initialized in Global Reset Mode or Channel Reset Mode

| Register | Bit / Flag |
|-------------------------------|---|
| RSCFDn(CFD)CmCTR register | (ROM), CRCT, CTMS[1:0], CTME, CHMDC[1:0] |
| RSCFDn(CFD)CmSTS register | CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, (ESIF), REC[7:0], TEC[7:0] |
| RSCFDn(CFD)CmERFL register | CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF |
| RSCFDn(CFD)CmFDCTR register | EOCCLR, SOCCLR |
| RSCFDn(CFD)CmFDSTS register | SOC[7:0], EOC[7:0], SOCO, EOCO, TDCVF, TDCR[6:0] |
| RSCFDn(CFD)CmFDCRC register | CRCREG[20:0], SCNT[3:0] |
| RSCFDn(CFD)CFCCk register | When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE |
| RSCFDn(CFD)CFSTSk register | When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF |
| RSCFDn(CFD)CFTISTS register | CFkTXIF |
| RSCFDn(CFD)TMCp register | TMOM, TMTAR, TMTR |
| RSCFDn(CFD)TMSTSp register | TMTARM, TMTRM, TMTRF[1:0], TMTSTS |
| RSCFDn(CFD)TMTRSTSy register | TMTRSTSp (Bits of corresponding channel are initialized in channel reset mode.) |
| RSCFDn(CFD)TMTARSTSy register | TMTARSTSp (Bits of corresponding channel are initialized in channel reset mode.) |
| RSCFDn(CFD)TMCSTSy register | TMCSTSp (Bits of corresponding channel are initialized in channel reset mode.) |
| RSCFDn(CFD)TMASTSy register | TMASTSp (Bits of corresponding channel are initialized in channel reset mode.) |
| RSCFDn(CFD)TXQCCm register | TXQE |
| RSCFDn(CFD)TXQSTSm register | TXQIF, TXQFLL, TXQEMP |
| RSCFDn(CFD)THLCCm register | THLE |
| RSCFDn(CFD)THLSTSm register | THLMC[4:0], THLIF, THLELT, THLFLL, THLEMP |
| RSCFDn(CFD)GTINTSTS0 register | TSIFm, TAIEm, TQIFm, CFTIFm, THIFm (m = 0 to 2) |

Note: Bits and flags in parentheses exist only in registers in CAN FD mode.

Table 17.181 Registers Initialized Only in Global Reset Mode

| Register | Bit / Flag |
|-----------------------------|--|
| RSCFDn(CFD)GSTS register | GHLTSTS |
| RSCFDn(CFD)GERFL register | EEF0, EEF1, EEF2, (CMPOF), THLES, MES, DEF |
| RSCFDn(CFD)GTSC register | TS[15:0] |
| RSCFDn(CFD)RMNDy register | RMNSq |
| RSCFDn(CFD)RFCCx register | RFE |
| RSCFDn(CFD)RFSTsx register | RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP |
| RSCFDn(CFD)CFCK register | When transmit/receive FIFO buffer is in receive mode: CFE |
| RSCFDn(CFD)CFSTSk register | When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT |
| RSCFDn(CFD)FESTS register | CFkEMP, RFxEMP |
| RSCFDn(CFD)FFSTS register | CFkFLL, RFxFLL |
| RSCFDn(CFD)FMSTS register | CFkMLT, RFxMLT |
| RSCFDn(CFD)RFISTS register | RFxIF |
| RSCFDn(CFD)CFRISTS register | CFkRXIF |
| RSCFDnCFDCTCT register | CFDMAEm, RFDMAEx |
| RSCFDnCFDCTSTS register | CFDMASTSm, RFDMASTx |
| RSCFDn(CFD)GTSTCFG register | RTMPS[6:0], C0ICBCE, C1ICBCE, C2ICBCE |
| RSCFDn(CFD)GTSTCTR register | RTME, ICBCTME |

Note: Bits and flags in parentheses exist only in registers in CAN FD mode.

17.7 Reception Functions

There are two reception types.

- Reception by receive buffers:
0 to 47 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of up to the number of buffer stages specified with the RFDC[2:0] and CFDC[2:0] bits can be stored in FIFO buffers and can be read sequentially from the oldest.

17.7.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows dispatching of selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 128 receive rules can be registered per channel and up to (64 × number of channels) total receive rules can be registered in the entire module. (Up to 192 receive rules can be registered in this module that has 3 channels.) Set receive rules for each channel. Receive rules cannot be shared with other channels. If receive rules are not set, no messages can be received. **Figure 17.6** illustrates how receive rules are registered.

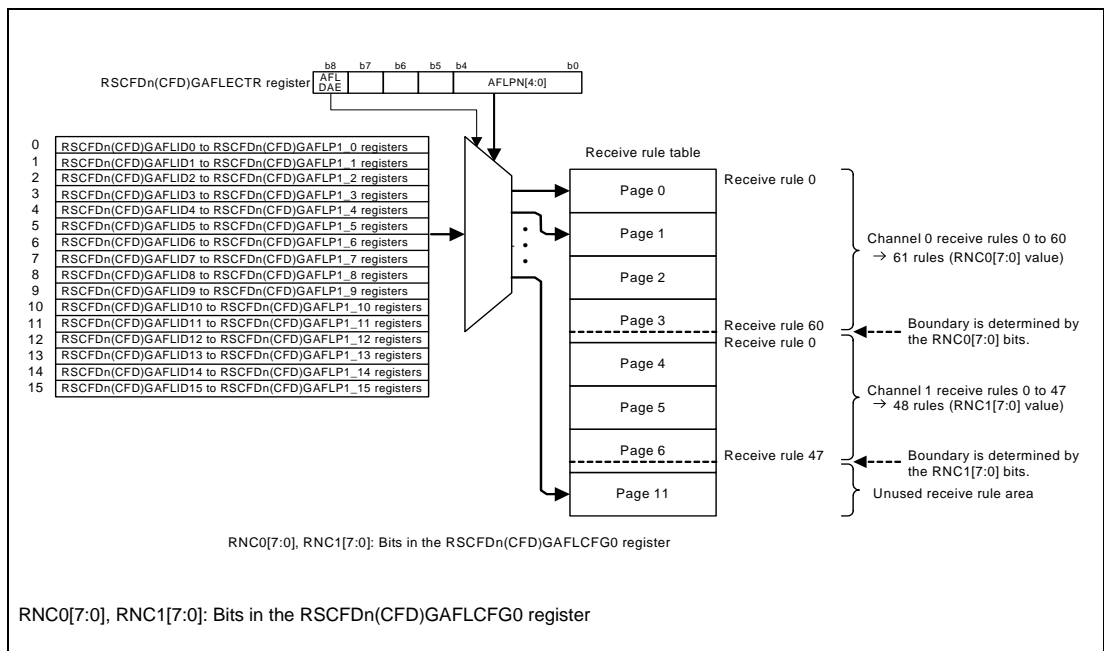


Figure 17.6 Registration of Receive Rules (for Setting Channel 0 and 1)

CAUTION

Receive rules for each channel must be set in contiguous blocks.
Channel 1 rules and channel 0 rules must be set separately.

Each receive rule consists of 16 bytes in the RSCFDn(CFD)GAFLIDj, RSCFDn(CFD)GAFLMj, RSCFDn(CFD)GAFLP0_j, and RSCFDn(CFD)GAFLP1_j registers (j = 0 to 15). The RSCFDn(CFD)GAFLIDj register is used to set GAFLID, GAFLIDE bit, GAFLRTR bit, and the mirror function, the RSCFDn(CFD)GAFLMj register is used to set mask, the RSCFDn(CFD)GAFLP0_j register is used to set label information to be added, DLC value, and storage receive buffer, and the RSCFDn(CFD)GAFLP1_j register is used to set storage FIFO buffer. Up to 16 receive rules can be set per page.

17.7.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in the received message which correspond to the bits set to 0 (bits are not compared) in the RSCFDn(CFD)GAFLMj register are not compared and are regarded as matched.

Check begins with the receive rule with the smallest number for the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

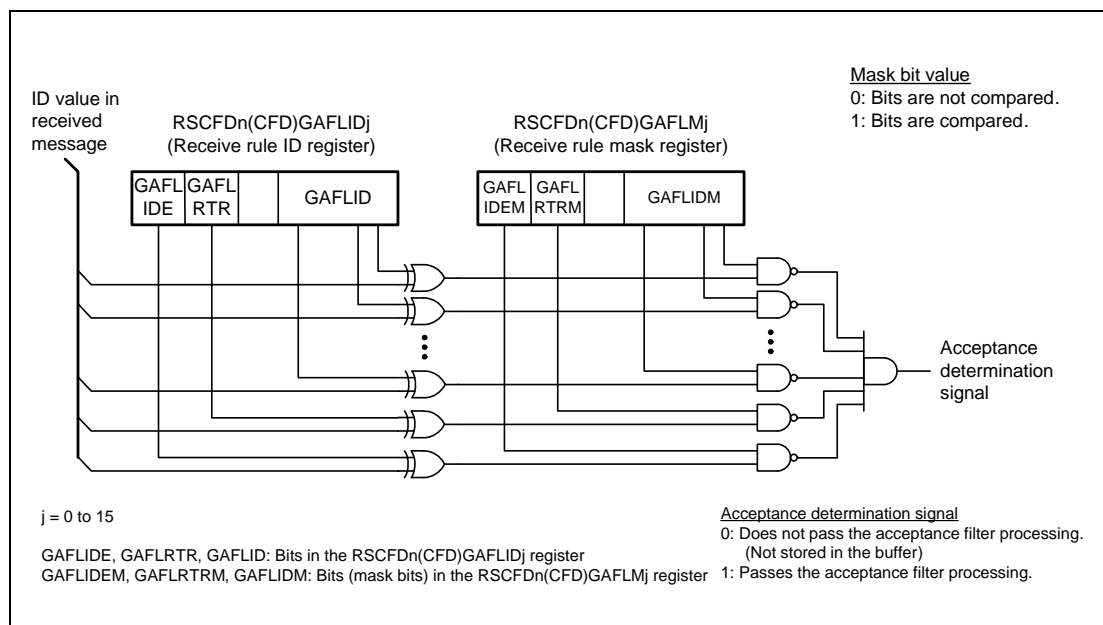


Figure 17.7 Acceptance Filter Function

17.7.1.2 DLC Filter Processing

When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), DLC filter processing is performed for messages that pass through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RSCFDn(CFD)GCFG register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RSCFDn(CFD)GCFG register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00_H is stored in each data byte beyond the number of bytes which is indicated by the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RSCFDn(CFD)GERFL register is set to 1 (a DLC error is present).

17.7.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, and/or transmit/receive FIFO buffers (set to receive mode or gateway mode). Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RSCFDn(CFD)GAFLP0_j register (j = 0 to 15) and by the RSCFDn(CFD)GAFLP1_j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to eight buffers.

In CAN FD mode, if the payload length of the received message exceeds the payload storage size of the storage buffer, the CMPOF flag in the RSCFDnCFDGERFL register is set to 1 (payload overflow) and the processing is handled according to the CMPOC bit in the RSCFDnCFDGCFG register. When the CMPOC bit is 0, the received message which exceeds the payload storage size is not stored in the buffer. When the CMPOC bit is 1, the received message is stored in the buffer with payloads exceeding the storage size being discarded, and depending on the DRE bit in the RSCFDnCFDGCFG register the received DLC value or the DLC value of the receive rule is stored in the buffer.

17.7.1.4 Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits in the RSCFDn(CFD)GAFLP0_j register.

17.7.1.5 Mirror Function Processing

The mirror function allows the CAN node to receive its own transmitted messages. The mirror function is enabled by setting the MME bit in the RSCFDn(CFD)GCFG register to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLLB bit in the RSCFDn(CFD)GAFLIDj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When the CAN node is receiving its own transmitted messages, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

17.7.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time and transmission time. The timestamp counter value is fetched at the timing set with the TSCCFG[1:0] bits in the RSCFDn(CFD)GCFG register and is then stored in a receive buffer or a FIFO buffer together with the message ID and data during data reception. The clock source of the timestamp counter is selected by the TSBTCS[2:0] and TSSS bits in the RSCFDn(CFD)GCFG register. In classical CAN mode, either $\text{pclk}/2$ or the CANm bit time clock ($m = 0$ to 2). In CAN FD mode, the clock source is selectable from $\text{pclk}/2$ or nominal CANm bit time clock. However, do not select the nominal CANm bit time clock of channels that handle CAN FD frames. The timestamp counter count source is obtained by dividing the selected clock source by the TSP[3:0] value in the RSCFDn(CFD)GCFG register.

When the CANm bit time clock or nominal CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the $\text{pclk}/2$ is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000_{H} by setting the TSRST bit in the RSCFDn(CFD)GCTR register to 1.

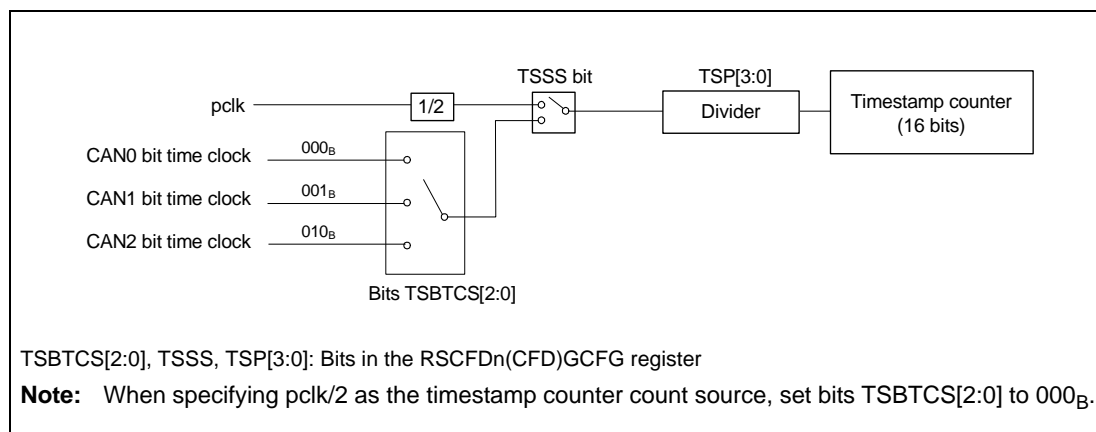


Figure 17.8 Timestamp Function Block Diagram

17.8 Transmission Functions

There are three types of transmission. In classical CAN mode, transmittable payload length is 8 bytes in every transmission type. In CAN FD mode, transmittable payload length varies with transmission types.

- Transmission using transmit buffers:
 Each channel has 16 buffers. Transmittable payload length in CAN FD mode is 20 bytes. However, when transmit buffer merge mode is used, four buffers out of 16 buffers are allocated as a payload-only storage area and two buffers are able to transmit payloads with a length of more than 20 bytes.
- Transmission using transmit/receive FIFO buffers (transmit mode):
 Each channel has three FIFO buffers. Up to 128 messages can be contained in a single FIFO buffer. Transmittable payload length in CAN FD mode is 64 bytes. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.
- Transmission using transmit queues:
 Up to 16 transmit buffers per channel can be allocated to the transmit queues. Transmittable payload length in CAN FD mode is 20 bytes. Transmit buffer $((16 \times m) + 15)$ is used as an access window of a corresponding channel. Transmit buffers are allocated to transmit queues in descending order of buffer number. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID number.

Figure 17.9 shows the allocation of transmit queues and transmit/receive FIFO buffer link.

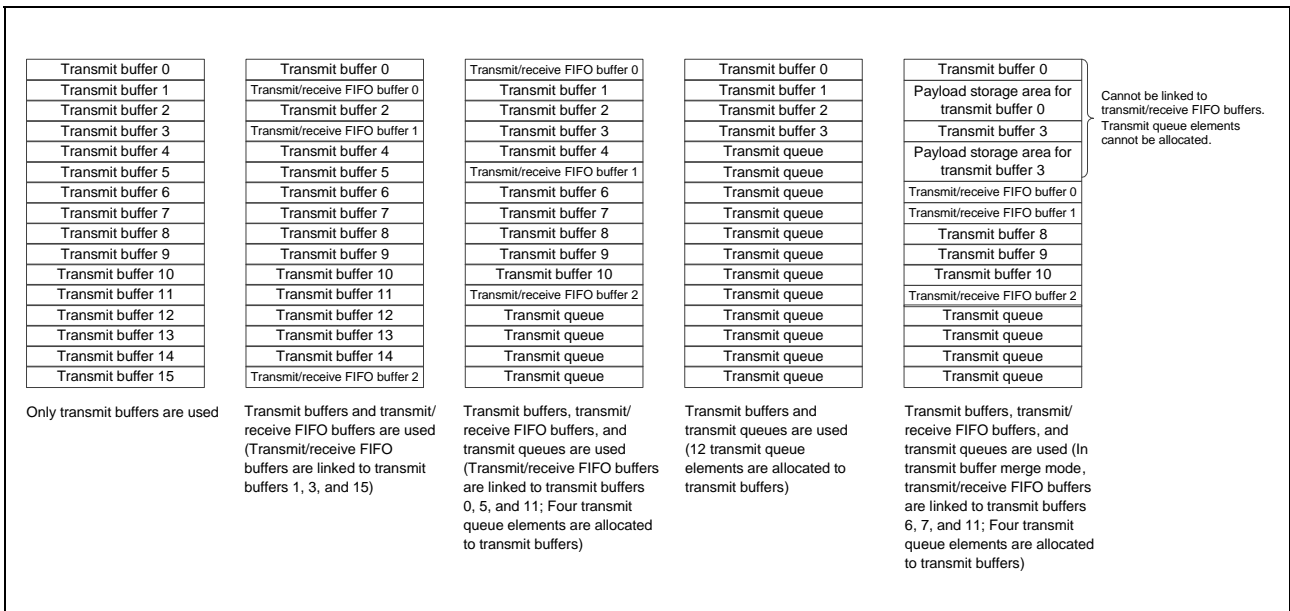


Figure 17.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

17.8.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or from the queue on the same channel, transmit priority is determined using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

All CAN channels use the setting of the TPRI bit in the RSCFDn(CFD)GCFG register.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. ID priority conforms to the CAN bus arbitration specification defined in the CAN specifications. All IDs of pending transmit messages are targets of priority determination, regardless of whether they are stored in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), or the transmit queue. If even a single transmit queue is used, select ID priority. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination.

When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When a transmit queue is used, all messages in the transmit queue are targets of priority determination. If the same ID is set for two or more buffers, the buffer with the smaller buffer number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer with the smallest buffer number among all buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again regardless of the TPRI bit setting. When a 2-bit ECC error is detected in the priority determination processing, no message is transmitted (only when the EEFEBIT in the RSCANnGCFG register is 1 in classical CAN mode).

17.8.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RSCFDn(CFD)TMCP register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

The transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCFDn(CFD)TMSTSp register ($p = 0$ to 47). When transmit completes successfully, the TMTRF[1:0] flag is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)).

17.8.2.1 Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the RSCFDn(CFD)TMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the RSCFDn(CFD)TMCP register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSp register is set to 01_B (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration-lost or an error occurs during transmission of a message for which the TMTAR bit is set to 1, retransmission is not performed.

17.8.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the RSCFDn(CFD)TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration-lost or an error occurs, retransmission is not performed.

The one-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCFDn(CFD)TMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10_B or 11_B. When an arbitration-lost or an error occurs, the TMTRF[1:0] flag is set to 01_B (transmit abort has been completed).

17.8.2.3 Transmit Buffer Merge Mode (Only in CAN FD Mode)

Transmit buffers can transmit messages with a payload length of 20 bytes, but can transmit messages with a payload length of up to 64 bytes by merging three transmit buffers in transmit buffer merge mode.

Setting the TMME bit to 1 in the RSCFDnCFDCmFDCFG register enables transmit buffer merge mode. In this mode, six buffers per channel become a merge area and two sets of transmit buffers $(16 \times m) + 0$ to $(16 \times m) + 2$ and transmit buffers $(16 \times m) + 3$ to $(16 \times m) + 5$ are merged. A transmission request is made by the first transmit buffer, and subsequent two buffers are used as a payload storage area. See **Section 17.11.3.1, Procedure for Transmission from Transmit Buffers** and **Table 17.188, Message Storage Registers in Transmit Buffer Merge Mode (Example of Transmit Buffer 0)**. Do not set the transmission request bit (TMTR bit in the RSCFDnCFDTMCp register) and the transmission abort request bit (TMTAR bit in the RSCFDnCFDTMCp register) to 1 for transmit buffers except for the first buffer.

While transmit buffer merge mode is enabled, do not link the transmit/receive FIFO buffer to six merged buffers or allocate it to the transmit queue.

17.8.3 Transmission Using FIFO Buffers

Multiple messages can be stored in a single transmit/receive FIFO buffer, up to the number specified by the FIFO buffer depth, which is set by the CFDC[2:0] bits in the RSCFDn(CFD)CFCCk register ($k = 0$ to 8). Messages are transmitted sequentially on a first-in, first-out basis.

Each transmit/receive FIFO buffer is linked to a transmit buffer selected by the CFTML[3:0] bits in the RSCFDn(CFD)CFCCk register. When the CFE bit in the RSCFDn(CFD)CFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority of only the next transmit message is determined in the FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately if the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, arbitration-lost or the transition to channel halt mode in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

17.8.3.1 Interval Transmission Function

A message transmission interval time can be set to space the transmission of messages from the same FIFO buffer when using a transmit/receive FIFO buffer set to transmit mode or gateway mode.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the RSCFDn(CFD)CFCCk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RSCFDn(CFD)CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H.

Select an interval timer count source using the CFITR and CFITSS bits in the RSCFDn(CFD)CFCCk register. When the CFITR and CFITSS bits are set to 00_B, the count source is obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10_B, the count source is obtained by dividing pclk/2 by (the value of the ITRCP[15:0] bits × 10). When the CFITR and CFITSS bits are set to x1_B, the CANm bit time clock becomes a count source in classical CAN mode and the nominal CANm bit time clock becomes a count source in CAN FD mode (Use this count source only for the channel does not handle the CAN FD frames in CAN FD mode).

The interval time is calculated by the following equations where M is the value of ITRCP[15:0] and N is the value of CFITT[7:0].

- When CFITR and CFITSS = 00_B:

$$\frac{1}{\text{pclk frequency}} \times 2 \times M \times N$$

- When CFITR and CFITSS = 10_B:

$$\frac{1}{\text{pclk frequency}} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS = x1_B:

$$\text{Classical CAN mode: } \frac{1}{\text{CANm bit time clock frequency}} \times N$$

$$\text{CAN FD mode: } \frac{1}{\text{Nominal CANm bit time clock frequency}} \times N$$

Figure 17.10 shows the interval timer block diagram.

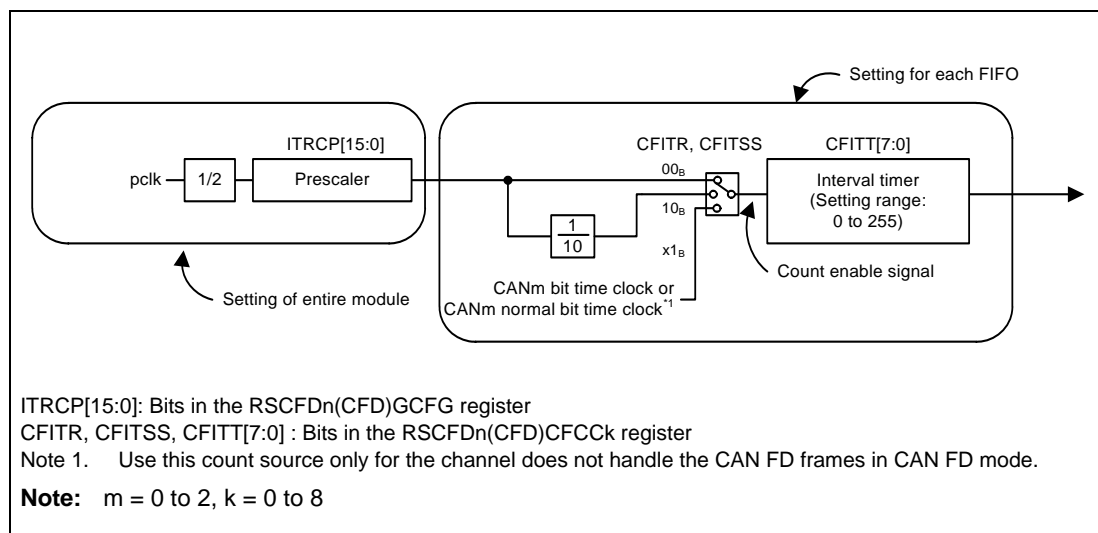


Figure 17.10 Interval Timer Block Diagram

Figure 17.11 shows the interval timer timing diagram.

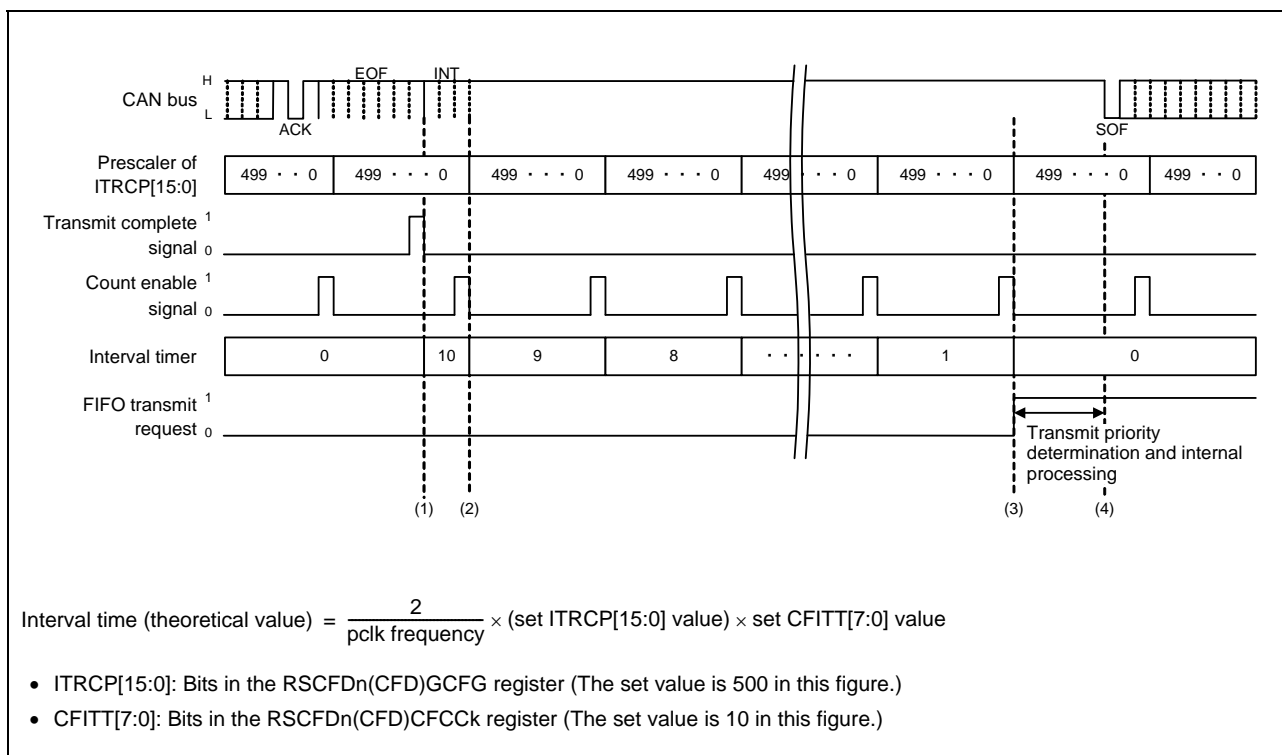


Figure 17.11 Interval Timer Timing Chart

- (1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.
- (2) The interval timer is decremented by 1 upon the next count enable signal.
- (3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.
- (4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request. If multiple internal processes (such as receive filter processing, message routing, and transmit priority determination) take place in all channels, a delay of up to 774 pclk cycles may be generated.

17.8.4 Transmission Using Transmit Queues

Three to sixteen buffers (up to 10 buffers in transmit buffer merge mode) are allocated to a transmit queue for each channel, and transmit buffer $((16 \times m) + 15)$ is used as an access window of a corresponding channel.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of storage sequence. If two messages having the same ID are stored in a transmit queue, these messages are not always transmitted in the order of their storage in the transmit queue.

Setting the TXQE bit in the RSCFDn(CFD)TXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RSCFDn(CFD)TXQSTSm register is set to 1 (the transmit queue contains no messages (transmit queue empty)) at the timing below.

- The transmit queue becomes empty immediately when no message in it is being transmitted or will be transmitted next.
- The transmit queue becomes empty after transmission completion, CAN bus error detection, arbitration-lost, or the transition to channel halt mode when a message in it is being transmitted or will be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

17.8.5 Transmit Data Padding (Only in CAN FD Mode)

When the payload length indicated by the set DLC value in a transmit message exceeds the payload storage area size of a buffer to be used for transmission, excessive payloads are padded by CC_H .

This processing is performed in the following cases when the transmit buffer merge mode is disabled (TMME bit in the RSCFDnCFDCmFDCFG register is 0).

- Transmit/receive FIFO set to transmission or gateway mode:
When the payload length of the transmit DLC exceeds the transmit/receive FIFO payload storage area size set by the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register
- Transmit buffer (including transmit queue):
When the payload length of the transmit DLC exceeds 20 bytes

When the transmit buffer merge mode is enabled, no transmit data is padded in any transmission using a transmit buffer, transmit/receive FIFO buffer, or transmit queue. At this time, do not set a payload length more than the payload storage size of the buffer for transmitting as the DLC value in the transmit message.

17.8.6 Transmit History Function

Information about transmission-completed messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 16 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RSCFDn(CFD)THLCCm register. The THLEN bit in the RSCFDn(CFD)CFIDk register ($k = 0$ to 8) determines whether transmit history data is stored for each message.]

In classical CAN mode, the TMTSCE bit in the RSCANnGCFG register can be used to set whether to include a timestamp value in the transmit history data. In CAN FD mode, a timestamp value is always included.

The following information on a transmitted message will be stored in the transmit history buffer after the successful completion of transmission.

Storage of the transmit history data after the successful completion of transmission may take up to 134 cycles of pclk in classical CAN mode or 402 pclk cycles in CAN FD mode.

- Buffer type 001_B: Transmit buffer
 010_B: Transmit/receive FIFO buffer
 100_B: Transmit queue
- Buffer number Number of source transmit buffer, transmit queue, or transmit/receive FIFO buffer. This number depends on buffer types. See **Table 17.182**.
- Label data Label information of the transmit message
- Timestamp Timestamp value of the transmit message
(When the TMTSCE bit is 1 in classical CAN mode)

Table 17.182 Transmit History Data Buffer Numbers

| Buffer No. \ Buffer type | 001 _B | 010 _B | 100 _B |
|--------------------------|-----------------------------|--|---|
| 0000 _B | Transmit buffer 16 × m + 0 | Buffer numbers of the transmit buffer linked to the transmit/receive FIFO buffer by the CFTML[3:0] bits in the RSCFDn(CFD)CFCK register (k = 0 to 8) | Buffer numbers of the transmit buffer allocated to the transmit queue that performed transmission |
| 0001 _B | Transmit buffer 16 × m + 1 | | |
| 0010 _B | Transmit buffer 16 × m + 2 | | |
| 0011 _B | Transmit buffer 16 × m + 3 | | |
| 0100 _B | Transmit buffer 16 × m + 4 | | |
| 0101 _B | Transmit buffer 16 × m + 5 | | |
| 0110 _B | Transmit buffer 16 × m + 6 | | |
| 0111 _B | Transmit buffer 16 × m + 7 | | |
| 1000 _B | Transmit buffer 16 × m + 8 | | |
| 1001 _B | Transmit buffer 16 × m + 9 | | |
| 1010 _B | Transmit buffer 16 × m + 10 | | |
| 1011 _B | Transmit buffer 16 × m + 11 | | |
| 1100 _B | Transmit buffer 16 × m + 12 | | |
| 1101 _B | Transmit buffer 16 × m + 13 | | |
| 1110 _B | Transmit buffer 16 × m + 14 | | |
| 1111 _B | Transmit buffer 16 × m + 15 | | |

Label data is used to identify each message. Unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

The timestamp value is fetched from the timestamp counter at the SOF (start of frame) timing of the message. For details about the timestamp counter, see **Section 17.7.1.6, Timestamp**.

Transmit history data can be read from the RSCFDn(CFD)THLACCm register. If an attempt is made to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

17.9 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, receive messages can be transmitted from an arbitrary channel without CPU intervention.

When the CFM[1:0] bits in the RSCFDn(CFD)CFCCk register are set to 10_B (gateway mode) for the transmit/receive FIFO buffer selected by the RSCFDn(CFD)GAFLP1_j register of a channel being used for transmission, messages that pass through filter processing according to the receive rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes the target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the RSCFDn(CFD)CFCCk register to 0 and the CFEMP flag becomes 1 according to the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted and will not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when the message in it is being transmitted or will be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

17.9.1 CAN-CAN FD Gateway (Only in CAN FD Mode)

When the gateway function is used in CAN FD mode, a frame to be transmitted can be replaced with a classical CAN frame or a CAN FD frame.

Setting the GWEN bit in the RSCFDnCFDCmFDCFG register to 1 enables the CAN-CAN FD gateway. The FDF and BRS bits in the transmit frame can be selected by the GWFDF and GWBRS bits in the RSCFDnCFDCmFDCFG register. When the DLC value of the received CAN frame is 1001_B or more and the GWFDF bit is 1 (CAN FD frame), the DLC value is replaced with 1000_B.

When the CAN-CAN FD gateway is enabled, do not perform routing for the following frames.

- CAN FD frames with a payload length of more than 8 bytes
- Remote frames

When the CAN-CAN FD gateway is enabled, the following frame should be transmitted in the channel by setting of GWFDF.

- When GWFDF bit is set to 0, only classical CAN frame should be transmitted.
- When GWFDF bit is set to 1, only CAN FD frame should be transmitted.

17.10 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
 - Standard test mode
 - Listen-only mode
 - Self-test mode 0 (external loopback mode)
 - Self-test mode 1 (internal loopback mode)
 - Restricted operation mode (only in CAN FD mode)
- Global tests: Performed for the entire module
 - RAM test (read/write test)
 - Inter-channel communication test [CRC error test enabled]

17.10.1 Standard Test Mode

CRC tests are enabled in standard test mode. The CRC value calculated by the RS-CANFD module based on the transmit message or receive message is stored in the register. This CRC value is stored in the CRCREG[14:0] bits in the RSCFDn(CFD)CmERFL register when the message is a classical CAN frame (CRC length = 15 bits) or in the CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register when the message is a CAN FD frame (CRC length = 17 or 21 bits). Use the inter-channel communication test function for CRC error tests. For details, see **Section 17.10.6.1, CRC Error Test**.

17.10.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

Figure 17.12 shows the connection when listen-only mode is selected.

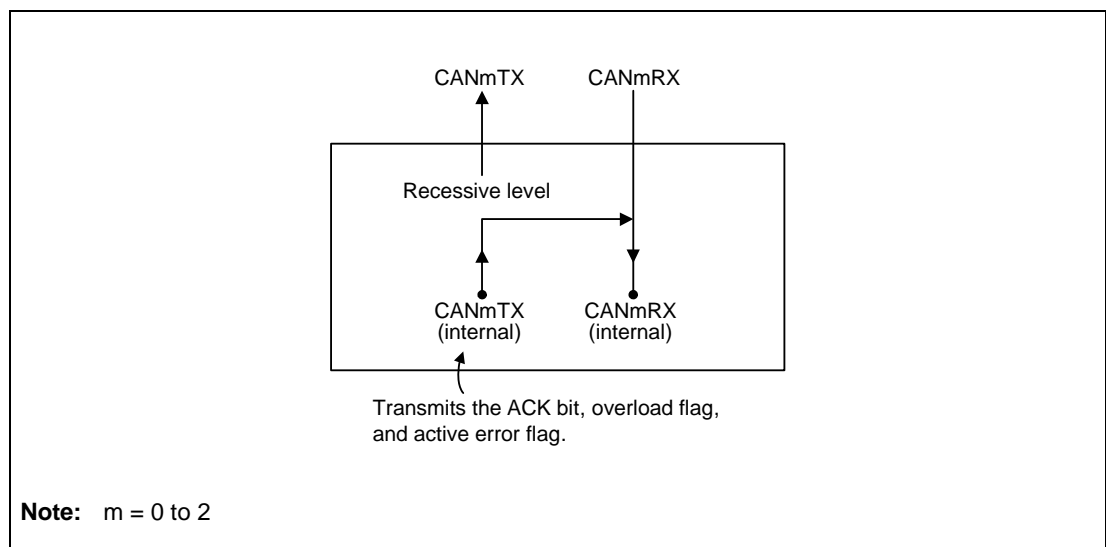


Figure 17.12 Connection when Listen-Only Mode is Selected

17.10.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RSCFDn(CFD)GAFLIDj register (j = 0 to 15) is set to 0 (when a message transmitted from another CAN node is received).

If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

17.10.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 17.13 shows the connection when self-test mode 0 is selected.

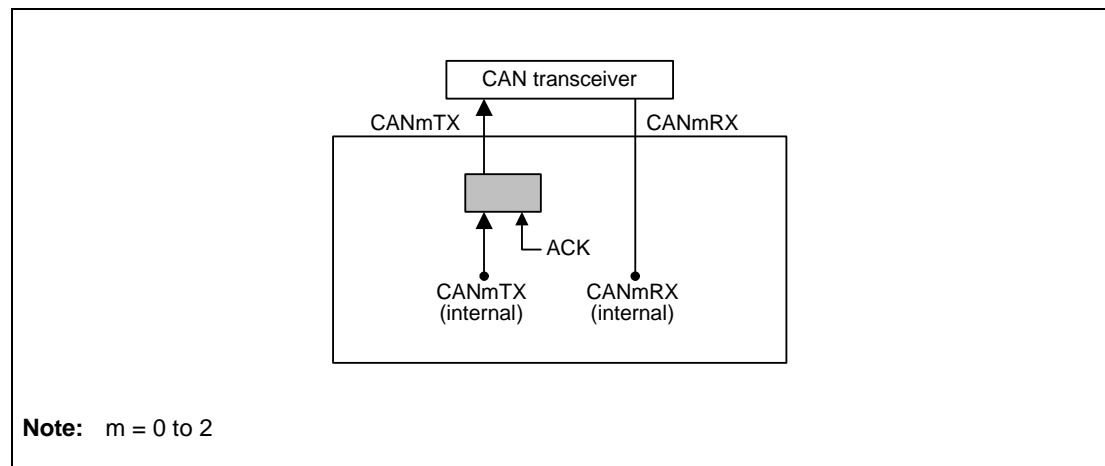


Figure 17.13 Connection when Self-Test Mode 0 is Selected

17.10.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANmTX pin ($m = 0$ to 2) to the internal CANmRX pin is performed. The external CANmRX pin input is isolated. The external CANmTX pin outputs only recessive bits.

Figure 17.14 shows the connection when self-test mode 1 is selected.

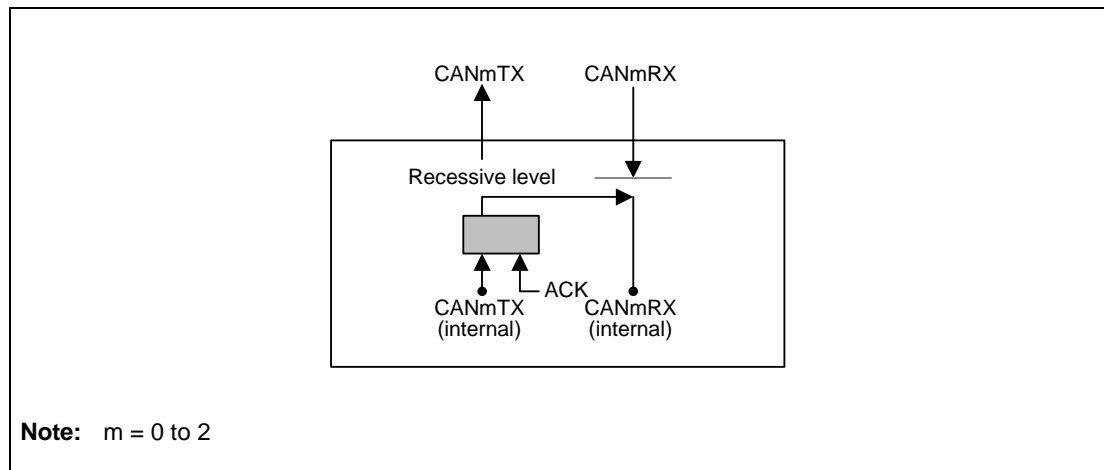


Figure 17.14 Connection when Self-Test Mode 1 is Selected

17.10.4 Restricted Operation Mode (Only in CAN FD Mode)

In restricted operation mode, an ACK bit is generated when a valid data frame and a remote frame have been received, but these frames are not transmitted even if an error frame or an overload frame transmit condition is detected. When a condition is detected, operation is suspended until the bus idle state comes for resynchronization with the CAN communication. The receive error counter (REC) and the transmit error counter (TEC) do not change due to an error.

A desired transmission request can be made for transmission without restrictions.

17.10.5 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test pages are set by the RTMPS[6:0] bits in the RSCFDn(CFD)GTSTCFG register. Data in the set page can be read from and written to the RSCFDn(CFD)RPGACC_r register ($r = 0$ to 63). The available total RAM size is 7680 bytes (1E00_H) in classical CAN mode or 10656 bytes (29A0_H) in CAN FD mode.

In CAN FD mode, do not access more than 160 bytes in the last page (RTMPS = 29_H) during RAM test.

17.10.6 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally connecting CAN channels to each other. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/reception settings for each channel. Set the channel(s) not participating in test to Channel halt mode.

Figure 17.15 shows the connection for inter-channel communication test.

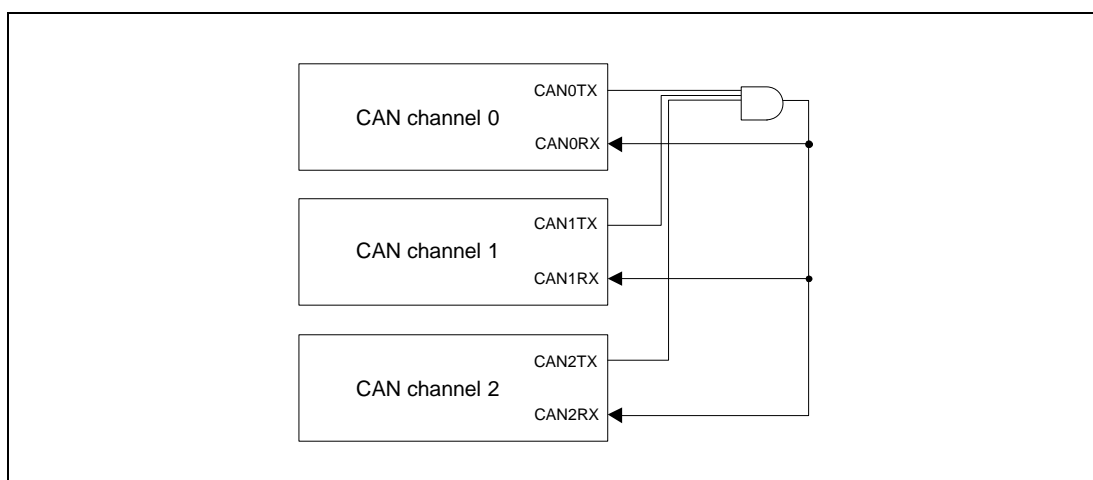


Figure 17.15 Connection for Inter-Channel Communication Test

17.10.6.1 CRC Error Test

A CRC error test is enabled during an inter-channel communication test. The following shows an example of channel 0 CRC error test procedure during a communication test between channel 0 and channel 1.

Preconditions

- Inter-channel communication test is enabled.
- Channel 0 and channel 1 are in standard test mode.

Procedure

1. Make a setting to send a message from the transmit buffer p of channel 1.
2. Set the CRCT bit in the RSCFDn(CFD)C0CTR register to 1 (to enable inversion of the first bit in the received ID field).
3. Set the TMTR bit in the RSCFDn(CFD)TMCp register to 1 (to issue a transmission request to the transmit buffer p of channel 1).
4. Wait for occurrence of a CAN0 error interrupt due to a channel bus error.
5. Read the CRCREG[14:0] bits in the RSCFDn(CFD)CmERFL register or the CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register of channel 0 and channel 1, and confirm that the CRC values are different on the transmission and the reception side.
6. Confirm that the CERR bit in RSCFDm(CFD)C0ERFL is 1 (CRC error detected).

The CRC error test function generates an incorrect CRC value by inverting the first bit in the received ID field. Therefore, note that not a CRC error but a stuff error (continuous 6-bit data of the same level) is detected when a message in which ID's upper 5-bit value is 10000_B or ID's upper 6-bit value is 011111_B is received.

The CRC generation circuit of the RS-CANFD module is contained in the protocol controller of each channel. Another CRC calculation test is not necessary during transmission because the same circuit is used for both transmission and reception.

17.11 RS-CANFD Setting Procedure

17.11.1 Initial Settings

The RS-CANFD module initializes the CAN RAM after the MCU is reset. The RAM initialization time is 3794 pclk cycles. The GRAMINIT flag in the RSCFDn(CFD)GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0. **Figure 17.16** shows the CAN setting procedure after the MCU is reset.

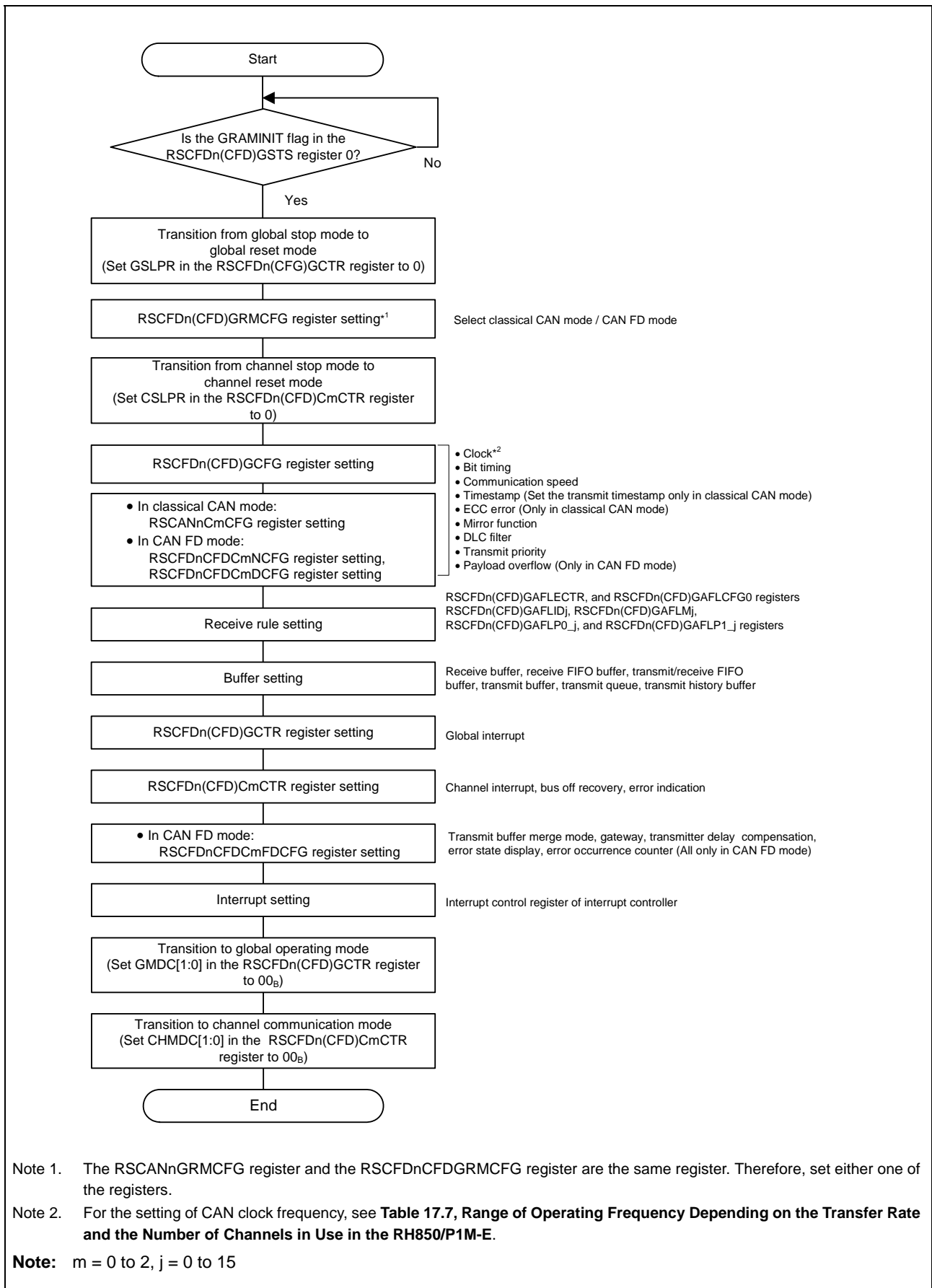


Figure 17.16 CAN Setting Procedure after the MCU is Reset

17.11.1.1 Clock Setting

Set the CAN clock (fCAN) as a clock source of the RS-CANFD module. Select the clk_xincan or clkc using the DCS bit in the RSCFDn(CFD)GCFG register.

17.11.1.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2, of which two segments TSEG1 and TSEG2 can be set by the corresponding registers for each channel. In classical CAN mode, set these two segments in the RSCANnCMCFG register. Two bit rates (nominal bit rate and data bit rate) are provided for CAN FD mode. Set the nominal bit rate in the RSCFDnCFDCmNCFG register and set the data bit rate in the RSCFDnCFDCmDCFG register. Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (hereafter referred to as Tq). A single Tq is the cycle of clock obtained by dividing the clock selected by the DCS bit in the RSCFDn(CFD)GCFG register. Set a division ratio by the BRP[9:0] bits in the RSCANnCMCFG register in classical CAN mode (CANmTq clock), and by the NBRP[9:0] bits in the RSCFDnCFDCmNCFG register and the DBRP[7:0] bits in the RSCFDnCFDCmDCFG register in CAN FD mode (CANmTq(N) clock and CANmTq(D) clock).

Be sure to specify the same value for both NBRP[9:0] and DBRP[7:0].

To specify different values for the nominal bit rate and the data bit rate, change the values of the RSCFDnCFDCmNCFG.NTSEG1 and NTSEG2 bits and RSCFDnCFDCmDCFG.DTSEG1 and DTSEG2 bits, respectively.

When the TDCE bit is set to 1 (Transmitter delay compensation is enabled) in the RSCFDnCFDCmDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

Figure 17.17 shows the bit timing chart. Table 17.183 shows an example of bit timing setting.

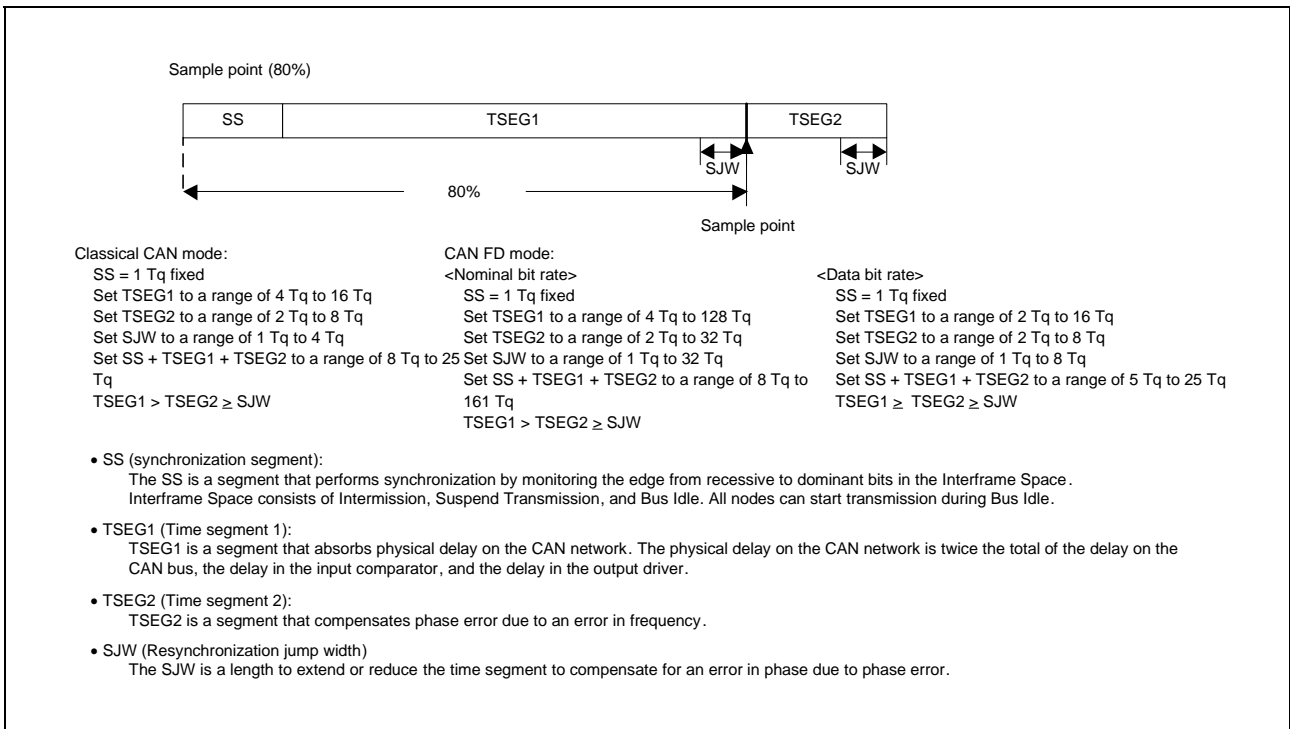


Figure 17.17 Bit Timing Chart

Table 17.183 Example of Bit Timing Settings

| 1 Bit | Set Value (Tq) | | | | Sample Point (%) Note: See Figure 17.17. |
|---------|----------------|-------|-------|-----|---|
| | SS | TSEG1 | TSEG2 | SJW | |
| 5 Tq*1 | 1 | 2 | 2 | 1 | 60.00 |
| 8Tq | 1 | 4 | 3 | 1 | 62.50 |
| | 1 | 5 | 2 | 1 | 75.00 |
| 10 Tq | 1 | 6 | 3 | 1 | 70.00 |
| | 1 | 7 | 2 | 1 | 80.00 |
| 16 Tq | 1 | 10 | 5 | 1 | 68.75 |
| | 1 | 11 | 4 | 1 | 75.00 |
| 20 Tq | 1 | 12 | 7 | 1 | 65.00 |
| | 1 | 13 | 6 | 1 | 70.00 |
| 50 Tq*1 | 1 | 39 | 10 | 4 | 80.00 |

Note 1. Only in CAN FD mode

17.11.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value, and Tq count per bit time. For CAN FD mode, set two types of transmission rate (arbitration phase and data phase) for each channel.

Figure 17.18 shows the CAN clock control block diagram, and **Table 17.184**, **Table 17.185** shows an example of the communication speed setting.

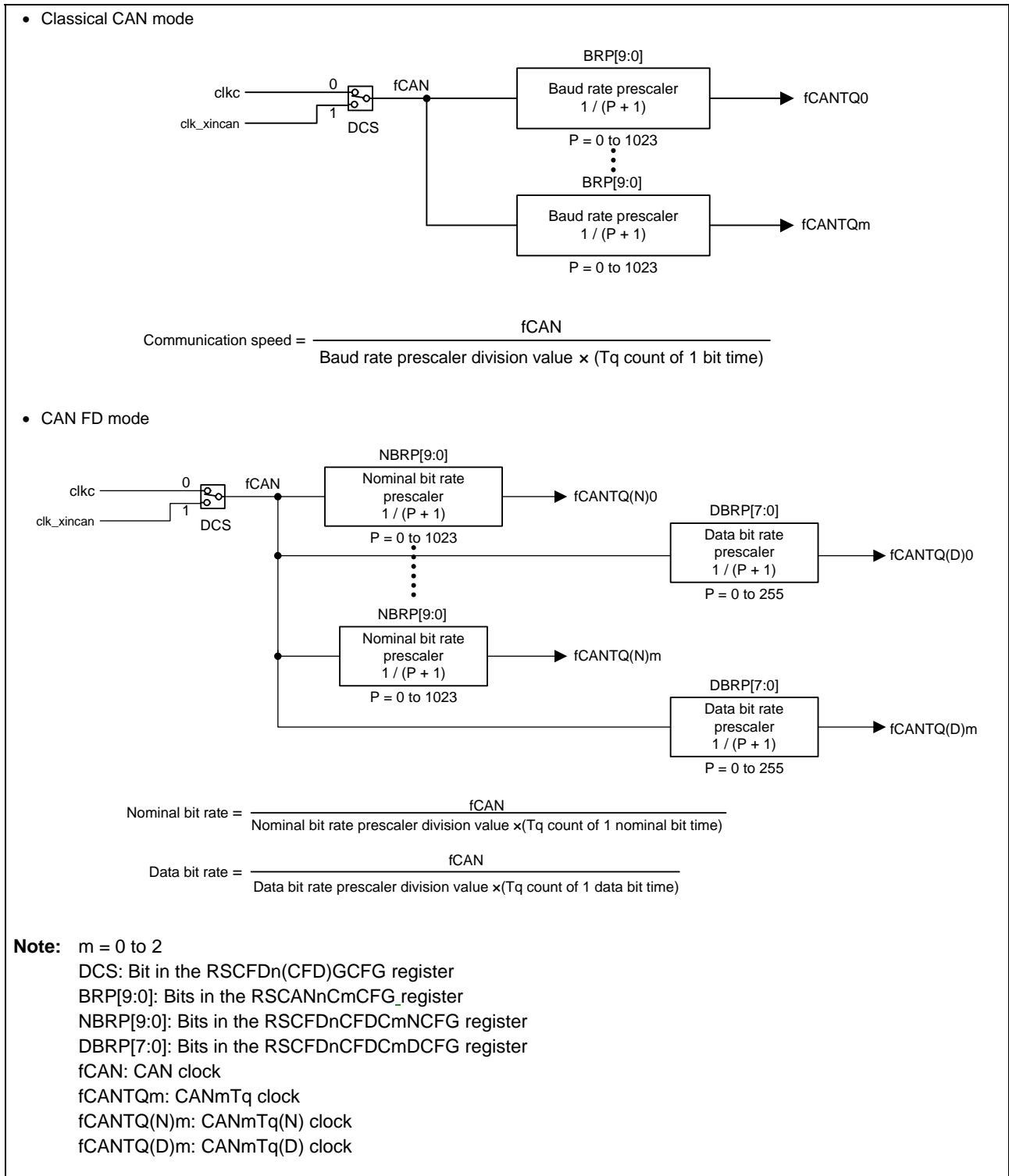


Figure 17.18 CAN Clock Control Block Diagram

Table 17.184 Example of Communication Speed Setting (Classical CAN mode)

| Communication Speed | fCAN | |
|---------------------|-------------------------|------------------------|
| | 40 MHz | 16 MHz |
| 1 Mbps | 8 Tq (5) 20 Tq (2) | 8 Tq (2) 16 Tq (1) |
| 500 Kbps | 8 Tq (10) 20 Tq (4) | 8 Tq (4) 16 Tq (2) |
| 250 Kbps | 8 Tq (20) 20 Tq (8) | 8 Tq (8) 16 Tq (4) |
| 125 Kbps | 8 Tq (40) 20 Tq (16) | 8 Tq (16) 16 Tq (8) |

Table 17.185 Example of Transmission Rate Setting (Nominal Bit Rate and Data Bit Rate in CAN FD Mode)

| Communication Rate | fCAN | |
|---|---|--|
| | 40 MHz | |
| Nominal bit rate 1 Mbps Data bit rate 5 Mbps | Nominal bit rate 40 Tq (1) Data bit rate 8 Tq (1) | |
| Nominal bit rate 500 kbps Data bit rate 2 Mbps | Nominal bit rate 80 Tq (1) Data bit rate 20 Tq (1) | |

Note: Values in () are baud rate prescaler division values.

17.11.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 11 by the AFLPN[4:0] bits in the RSCFDn(CFD)GAFLECTR register. Set receive rule table write enable/disable using the AFLDAE bit.

Figure 17.19 shows the receive rule setting procedure.

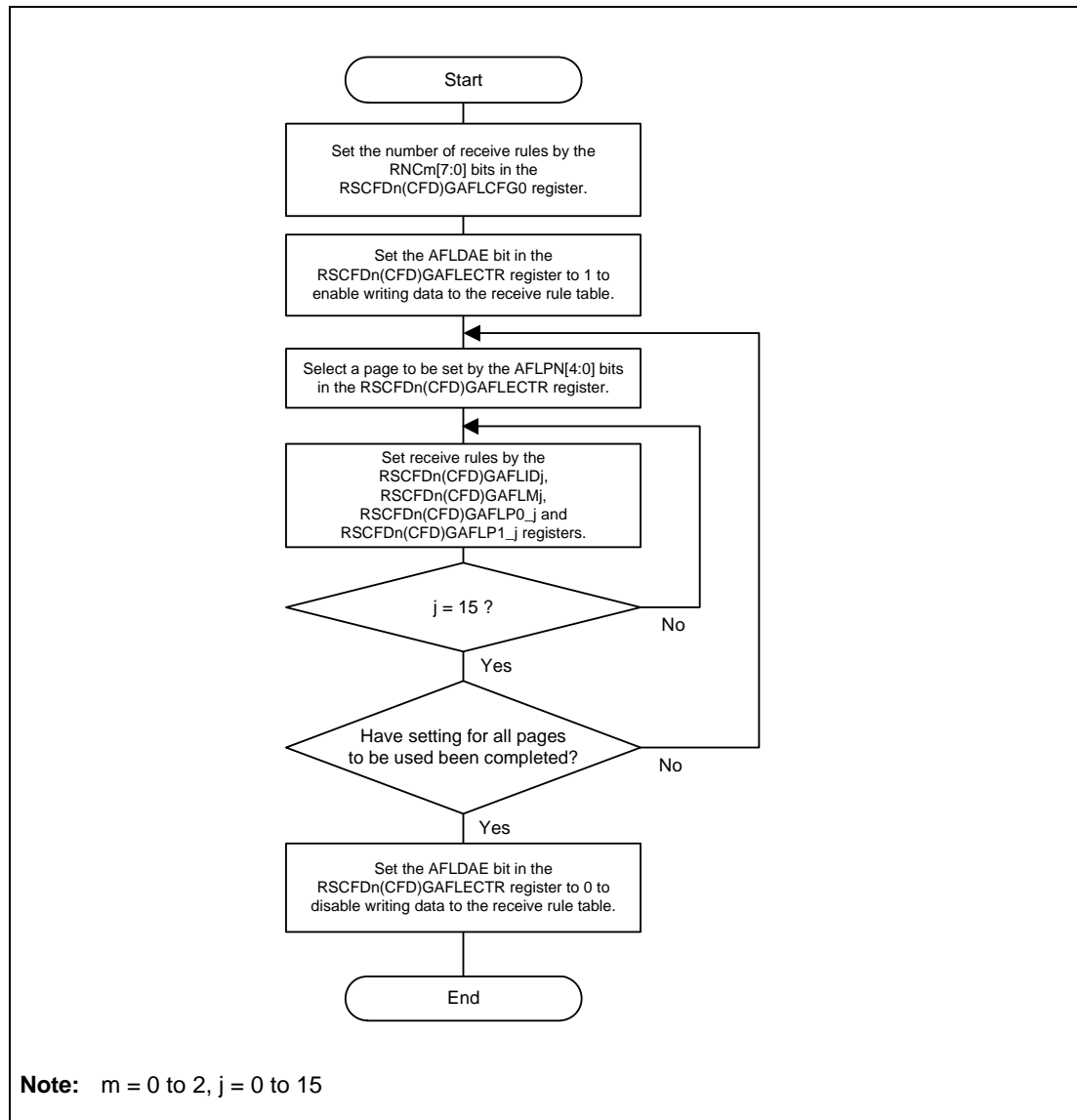


Figure 17.19 Receive Rule Setting Procedure

17.11.1.5 Buffer Setting

Set the number of buffers to be used (number of messages to be stored) and interrupt sources. Also set the payload storage size for CAN FD mode. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

In classical CAN mode, up to 3072 bytes of the RAM can be used in receive buffers and FIFO buffers. Up to 192 buffers are available, and 16 bytes are used per buffer. Configure the buffers so that the following conditions are met.

$$\text{Number of receive buffers} + \text{total number of depth of receive FIFO buffers } x + \text{total number of depth of transmit/receive FIFO buffers } k \leq 192 \text{ buffers}$$

In CAN FD mode, up to 5376 bytes of the RAM can be used in receive buffers and FIFO buffers. Configure the buffers so that the following conditions are met.

$$\text{Number of receive buffers} \times (12 + \text{payload storage size}) + \text{total of (number of depth} \times (12 + \text{payload storage size})) \text{ of receive FIFO buffers } x + \text{total of (number of depth} \times (12 + \text{payload storage size})) \text{ of transmit/receive FIFO buffers } k \leq 5376 \text{ bytes}$$

Figure 17.20 shows the buffer configuration. **Figure 17.21** shows the buffer setting procedure.

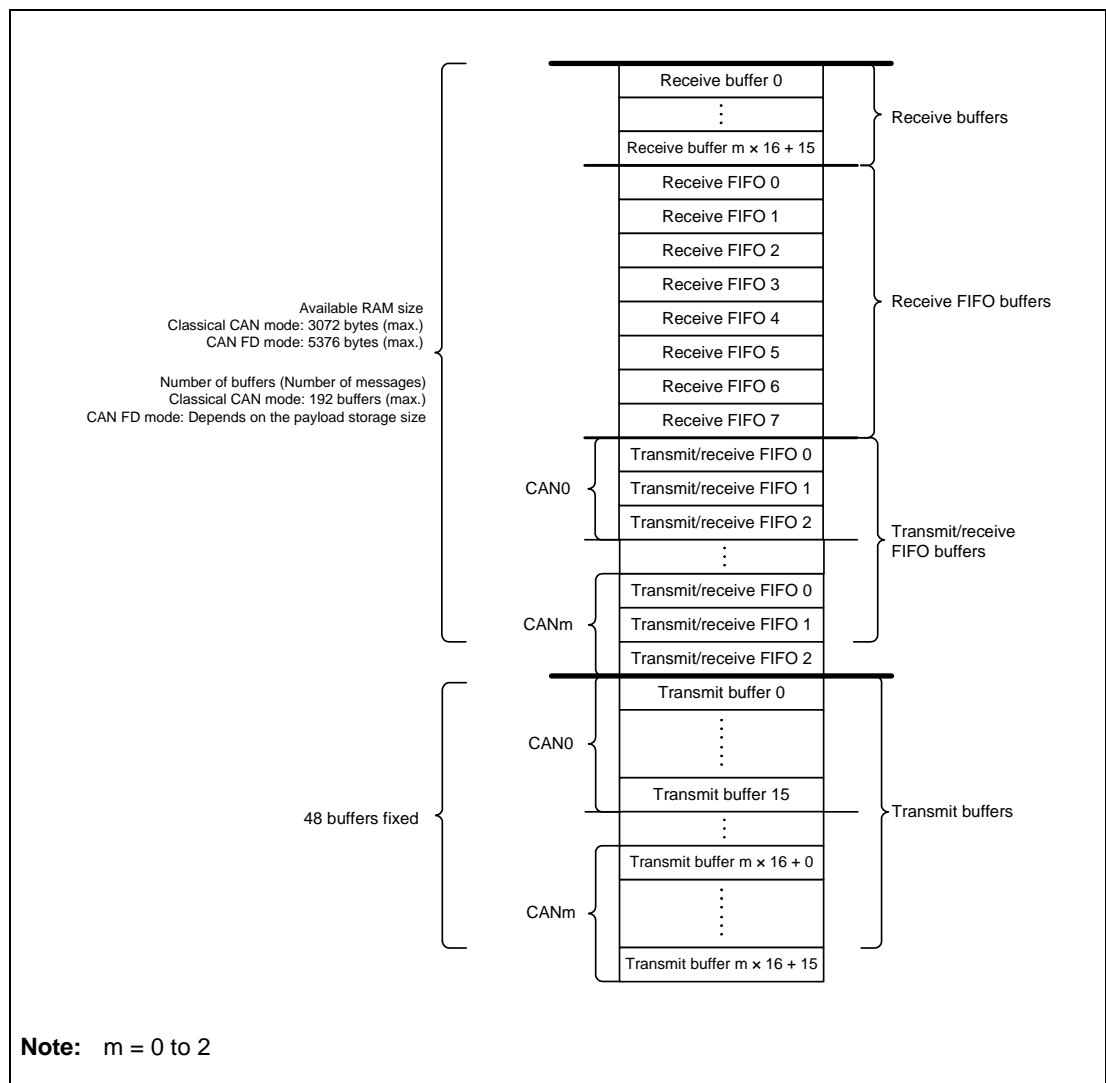


Figure 17.20 Buffer Configuration

CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.

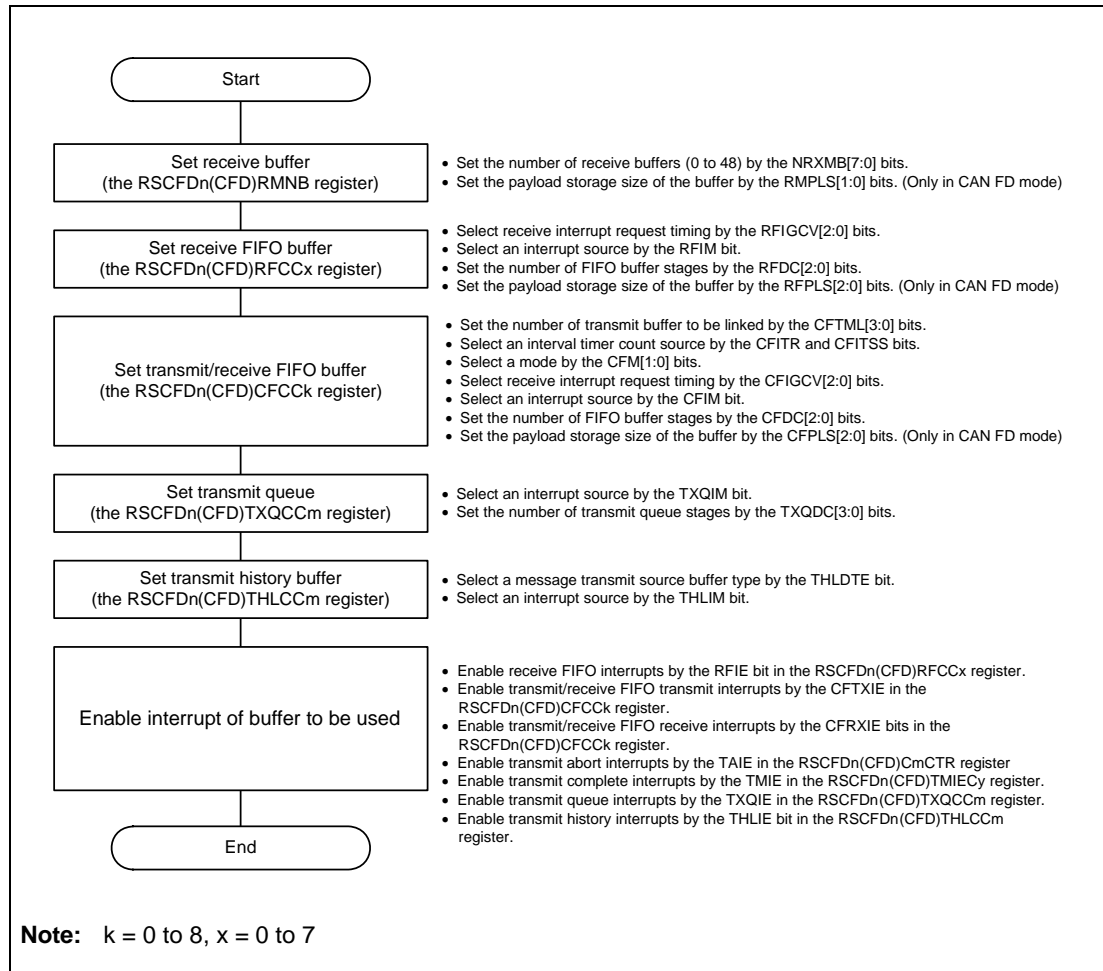


Figure 17.21 Buffer Setting Procedure

17.11.1.6 Transmitter Delay Compensation (Only in CAN FD Mode)

A high baud rate is used in the data phase in CAN FD mode. Transmitter delay compensation is provided as a function to accept propagation delay in this case.

To use this function, set the TDCE bit in the RSCFDnCFDCmFDCFG register to 1. Also set the secondary sample point (SSP) timing used in the data phase by the TDCOC bit and TDCO[6:0] bits in the RSCFDnCFDCmFDCFG register.

When the TDCOC bit is 0, the SSP timing equals the total value of the delay measured by the RS-CANFD module and the TDCO[6:0] value. (This value is rounded off to the nearest integer of T_q .) Usually, the TDCO[6:0] value must be equal to SS + TSEG1, the sample point timing

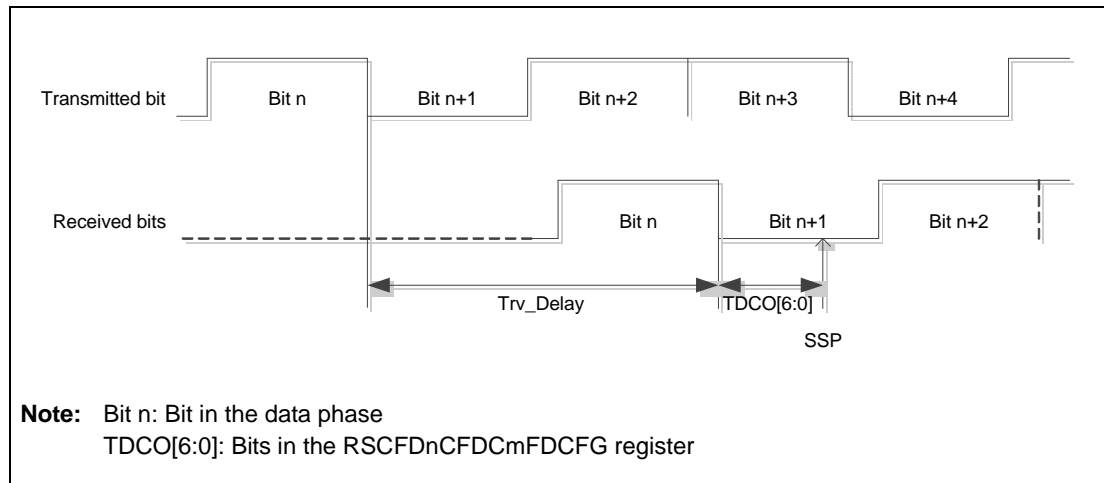


Figure 17.22 SSP timing

When the TDCOC bit is 1, the SSP timing is determined only by the TDCO[6:0] value. (When the DBRP[7:0] value in the RSCFDnCFDCmDCFG register is larger than 0, the TDCO[6:0] value is also rounded off to the nearest integer of T_q .)

The RS-CANFD module compensates a delay up to $(3 \text{ CANm bit time} - 2 f_{\text{CAN}})$. (CANm bit time is data bit rate value.)

When the TDCE bit is set to 1 (Transmitter delay compensation is enabled) in the RSCFDnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

17.11.2 Reception Procedure

17.11.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RSCFDn(CFD)RMNDy register ($y = 0, 1, q = 0$ to 47) is set to 1 (receive buffer q contains a new message). Messages can be read from registers RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDn(CFDRMFDSTSq (only in CAN FD mode), and RSCFDn(CFD)RMDFb_q ($b = 0$ or 1 in classical CAN mode, $b = 0$ to 4 in CAN FD mode). **Figure 17.23** shows the receive buffer reading procedure. This procedure ensures the consistency of messages read from registers RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDn(CFDRMFDSTSq, and RSCFDn(CFD)RMDFb_q.

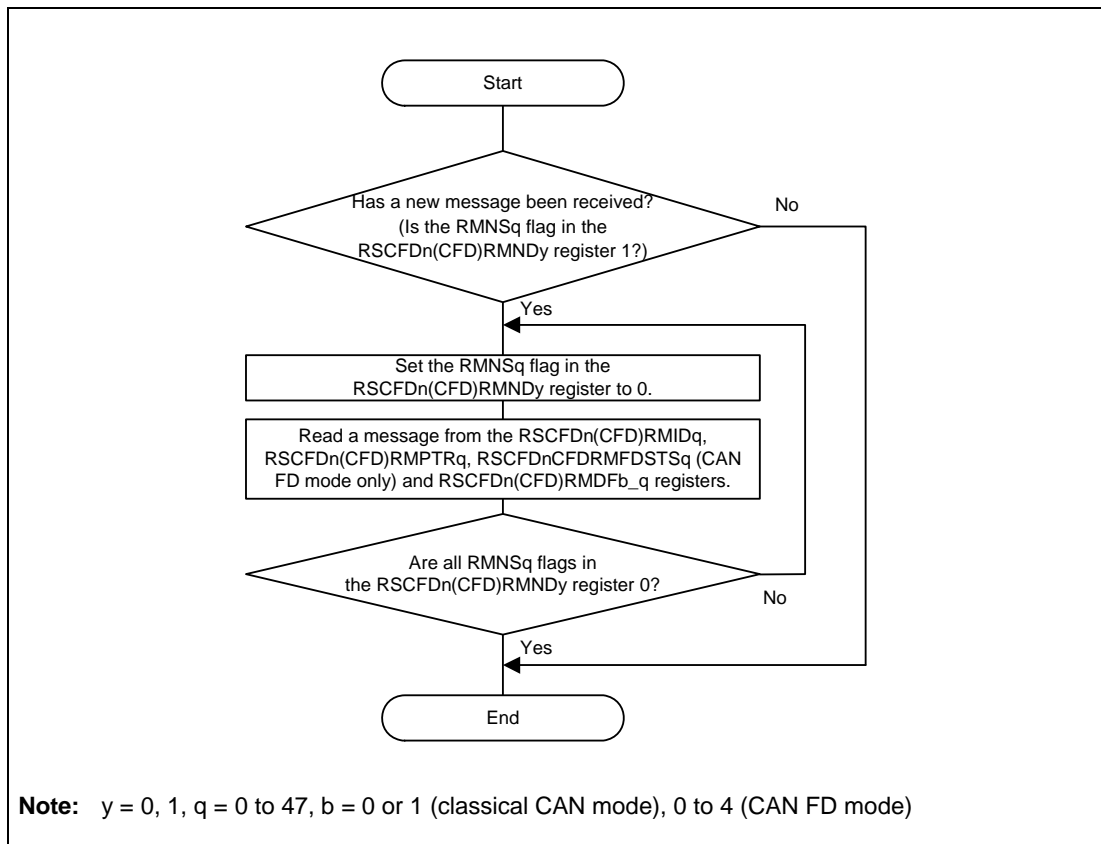


Figure 17.23 Receive Buffer Reading Procedure

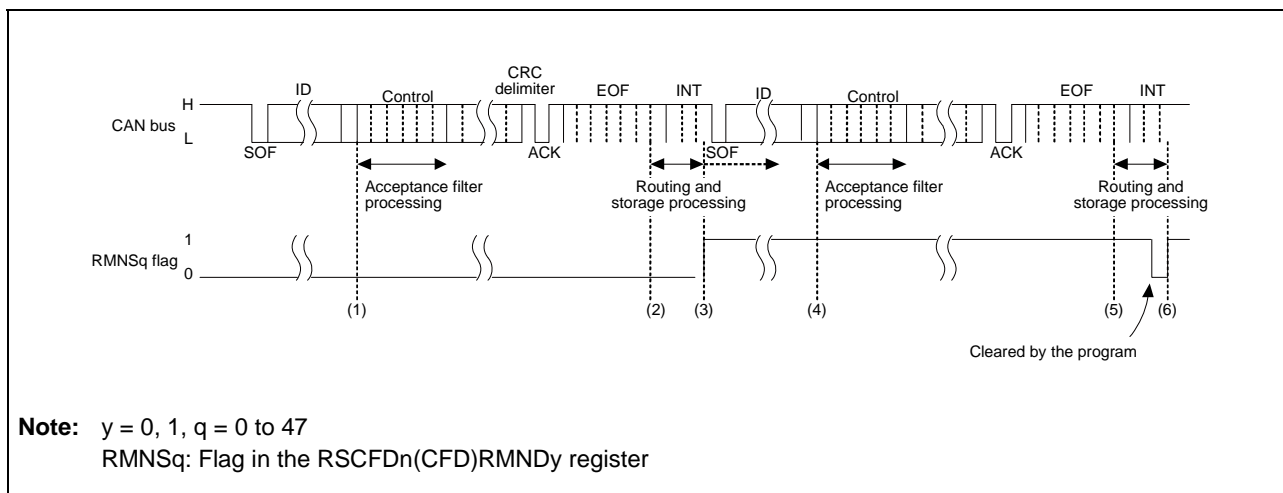


Figure 17.24 Receive Buffer Reception Timing Chart

- (1) When the ID field in a message has been received, the acceptance filter processing starts.
- (2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the RMNSq flag in the corresponding RSCFDn(CFD)RMNDy register is set to 1 (the receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.
- (4) When the ID field of the next message has been received, the acceptance filter processing starts.
- (5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (6) When the corresponding RMNSq flag is cleared to 0 (the receive buffer contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag cannot be cleared to 0 during storage of messages.

17.11.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode or gateway mode, the corresponding message counter (RFMC[7:0] bits in the RSCFDn(CFD)RFSTSx register (x = 0 to 7) or CFMC[7:0] bits in the RSCFDn(CFD)CFSTS_k register (k = 0 to 8)) is incremented by 1. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RSCFDn(CFD)RFCCx register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RSCFDn(CFD)CFCCk register is set to 1, an interrupt request is generated. Received messages can be read from the RSCFDn(CFD)RFID_x, RSCFDn(CFD)RFPTR_x, RSCFDn(CFD)RFFDSTS_x (only in CAN FD mode), and RSCFDn(CFD)RFDFd_x (d = 0 or 1 in classical CAN mode, d = 0 to 15 in CAN FD mode) registers for receive FIFO buffers, or from the RSCFDn(CFD)CFID_k, RSCFDn(CFD)CFPTR_k, RSCFDn(CFD)CFDCSTS_k (only in CAN FD mode), and RSCFDn(CFD)CFDFd_k registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RSCFDn(CFD)RFCCx register or the CFDC[2:0] bits in the RSCFDn(CFD)CFCCk register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RSCFDn(CFD)RFSTSx register or the CFEMP flag in the RSCFDn(CFD)CFSTS_k register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RSCFDn(CFD)RFSTSx register or CFRXIF flag in the RSCFDn(CFD)CFSTS_k register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.

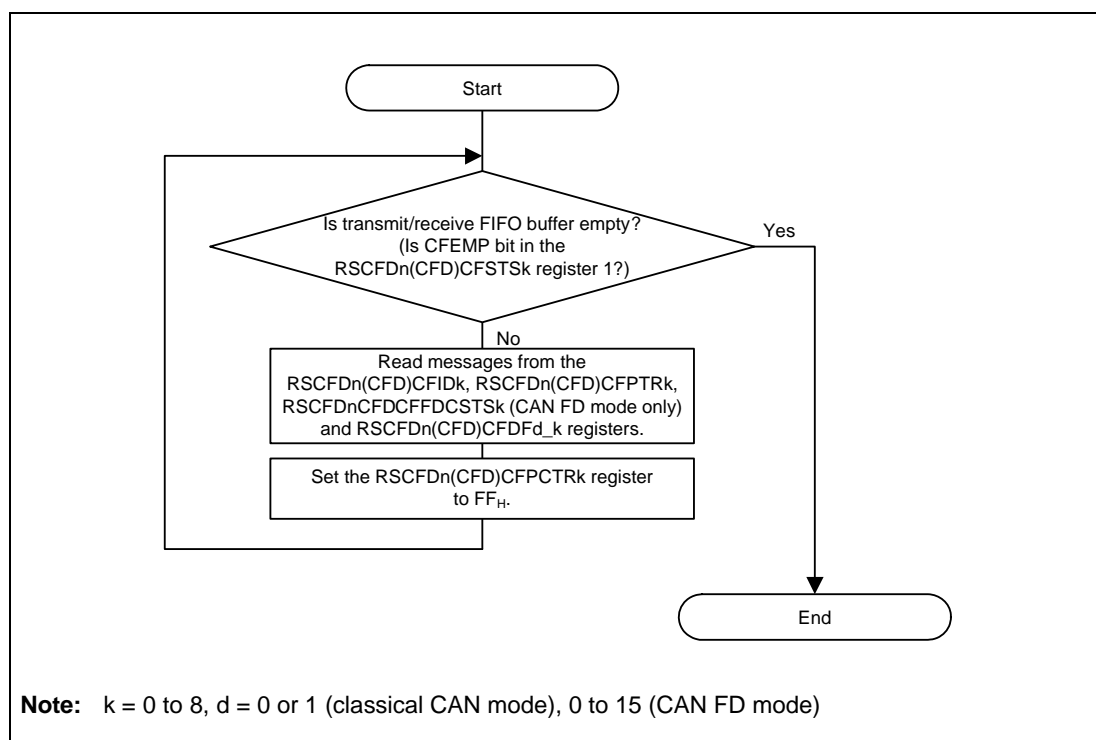


Figure 17.25 Transmit/Receive FIFO Buffer Reading Procedure

When reading a message in CAN FD mode, do not read the RSCFDnCFDRFDFd_x or RSCFDnCFDCFDf_k register corresponding to the area exceeding the payload storage size specified by the RFPLS[2:0] bits in the RSCFDnCFDRFCCx register or the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register.

Table 17.186 Payload Storage Area of Receive FIFO Buffer

| Set RFPLS[2:0] Value | Payload Storage Size | Corresponding Data Field Registers |
|----------------------|----------------------|---------------------------------------|
| 000 _B | 8 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF1_x |
| 001 _B | 12 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF2_x |
| 010 _B | 16 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF3_x |
| 011 _B | 20 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF4_x |
| 100 _B | 24 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF5_x |
| 101 _B | 32 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF7_x |
| 110 _B | 48 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF11_x |
| 111 _B | 64 bytes | RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF15_x |

Table 17.187 Payload Storage Area of Transmit/Receive FIFO Buffer

| Set CFPLS[2:0] Value | Payload Storage Size | Corresponding Data Field Registers |
|----------------------|----------------------|---------------------------------------|
| 000 _B | 8 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf1_k |
| 001 _B | 12 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf2_k |
| 010 _B | 16 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf3_k |
| 011 _B | 20 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf4_k |
| 100 _B | 24 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf5_k |
| 101 _B | 32 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf7_k |
| 110 _B | 48 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf11_k |
| 111 _B | 64 bytes | RSCFDnCFDCFDf0_k to RSCFDnCFDCFDf15_k |

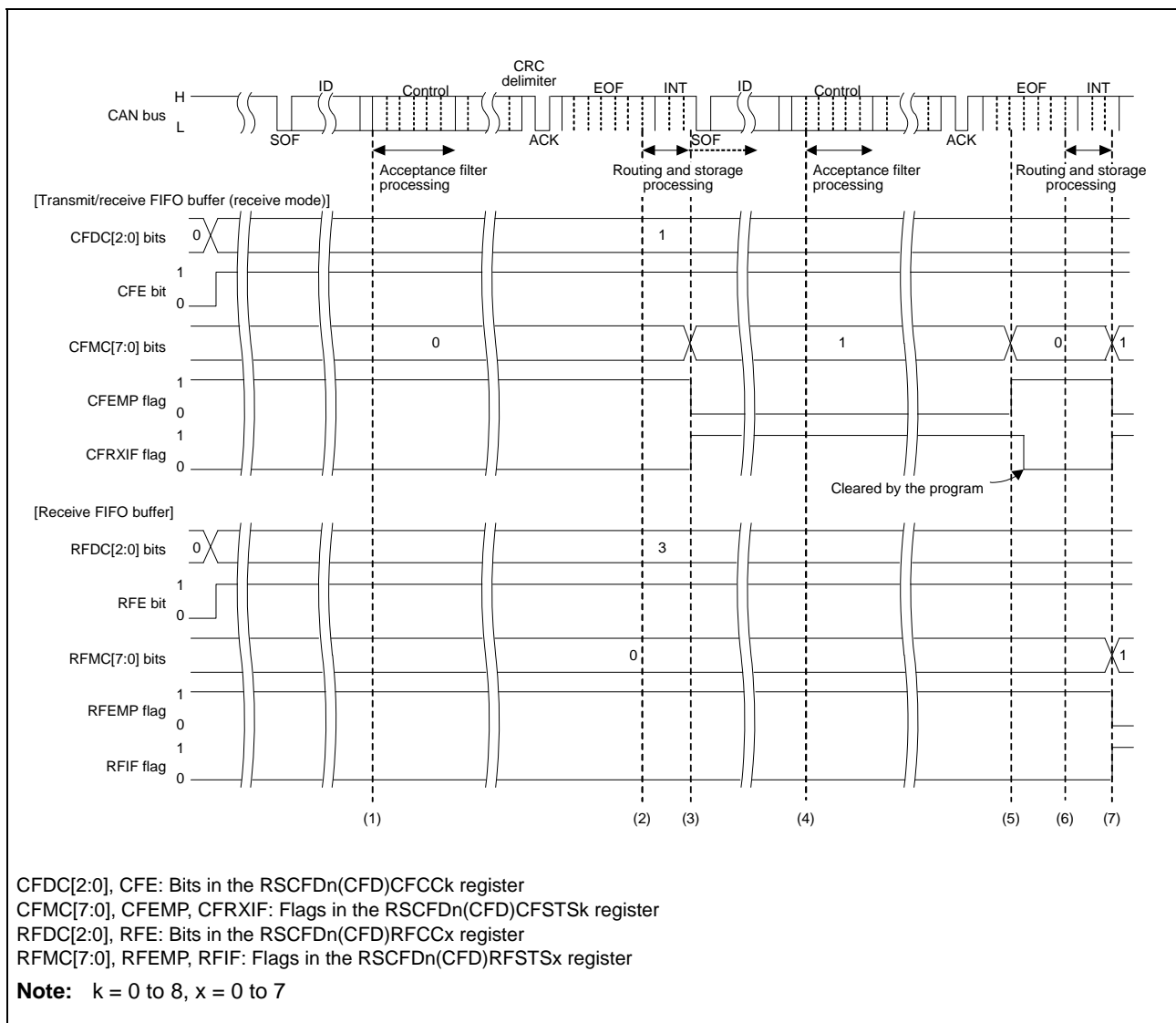


Figure 17.26 FIFO Buffer Reception Timing Chart

- (1) When the ID field in a message has been received, the acceptance filter processing starts.
- (2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (3) When the message has passed through the DLC filter processing and the CFE bit in the RSCFDn(CFD)CFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCFDn(CFD)CFCCk register is 001_B or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] value in the RSCFDn(CFD)CFSTSk register is incremented and becomes 01_H. When the CFIM bit in the RSCFDn(CFD)CFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RSCFDn(CFD)CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.
- (4) When the ID field of the next message has been received, the acceptance filter processing starts.

- (5) Read received messages from the RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, and RSCFDn(CFD)CFDFd_k registers and write FF_H to the RSCFDn(CFD)CFPCTRk register. This causes the CFMC[7:0] bits in the RSCFDn(CFD)CFSTSk register to be decremented. When CFMC[7:0] becomes 00_H, the CFEMP flag in the RSCFDn(CFD)CFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).
- (6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (7) The message is stored in the transmit/receive FIFO buffer set in receive mode when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used), and the CFDC[2:0] bits are set to 001_B or more. The CFMC[7:0] bit value is incremented by 1 to be 01_H. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer if the RFE bit in the RSCFDn(CFD)RFCCx register is set to 1 (receive FIFO buffers are used), and the RFDC[2:0] bits in the RSCFDn(CFD)RFCCx register are set to 001_B or more. The RFMC[7:0] bits in the RSCFDn(CFD)RFSTsx register are set to 01_H by being incremented by 1. When the RFIM bit in the RSCFDn(CFD)RFCCx register is set to 1 (an interrupt request occurs each time a message has been received), the RFIF flag in the RSCFDn(CFD)RFSTsx register is set to 1 (a receive FIFO interrupt request is present).

17.11.2.3 FIFO Buffer Reading Procedure by DMA Transfer

In CAN FD mode, the following FIFO buffers can be read by DMA transfer.

- All receive FIFO buffers x ($x = 0$ to 7)
- The first transmit/receive FIFO buffer k allocated to channel m ($k = 3 \times m$, $m = 0$ to 2)

The DMA enable bit (RFDMAEx or CFDMAEm bit in the RSCFDnCFDCDTCT register) can be set at any time. However, before setting the DMA enable bit to 1 (to enable DMA transfer requests), set the receive interrupt enable bit (RFIE bit in the RSCFDnCFDRFCCx register or CFRXIE bit in the RSCFDnCFDCFCCK register) of related FIFO buffers to 0 (to disable interrupts). When DMA transfer requests are enabled, do not write a value to the FIFO control register (RSCFDnCFDRFCCx register or RSCFDnCFDCFCCK register).

When an unread message is remaining in a DMA transfer-enabled FIFO buffer, a DMA transfer request trigger is generated. Specify the FIFO access register address(*) for the transfer source address, and adjust the transfer size so that data can be read to the end of the payload storage area with a single trigger. The end of the payload storage area depends on the payload storage size specified by the RFPLS[2:0] bits in the RSCFDnCFDRFCCx register or the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register.

After the end of the payload stored in the FIFO buffer has been read, the RFMC[7:0] value in the RSCFDnCFDRFSTSx register or the CFMC[7:0] value in the RSCFDnCFDCFCSTSx register is automatically decremented by 1. After that, if an unread message is remaining in the FIFO buffer, a trigger is generated again.

When the RFDMAEx or CFDMAEm bit is set to 0 (to disable DMA transfer requests) during DMA transfer, wait until the DMA transfer status (RFDMASTSx or CFDMASTSx bit in the RSCFDnCFDCDTSTS register) is cleared to 0 (DMA transfer disabled), and then start the next processing (enabling DMA transfer again etc.). When disabling DMA transfer, examine how to process a message remaining in the FIFO buffer and a newly arriving message. When the FIFO buffer is enabled, it continues to receive messages.

***Note1:**

- **Receive FIFO buffer**
RSCFDnCFDRFIDx, RSCFDnCFDRFPTRx, RSCFDnCFDRFFDSTSx,
RSCFDnCFDRFDFd_x
- **Transmit/Receive FIFO buffer**
RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSx,
RSCFDnCFDCFDFd_k

17.11.3 Transmission Procedure

17.11.3.1 Procedure for Transmission from Transmit Buffers

Figure 17.27 shows the procedure for transmission from transmit buffers.

Figure 17.28 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 17.29** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.

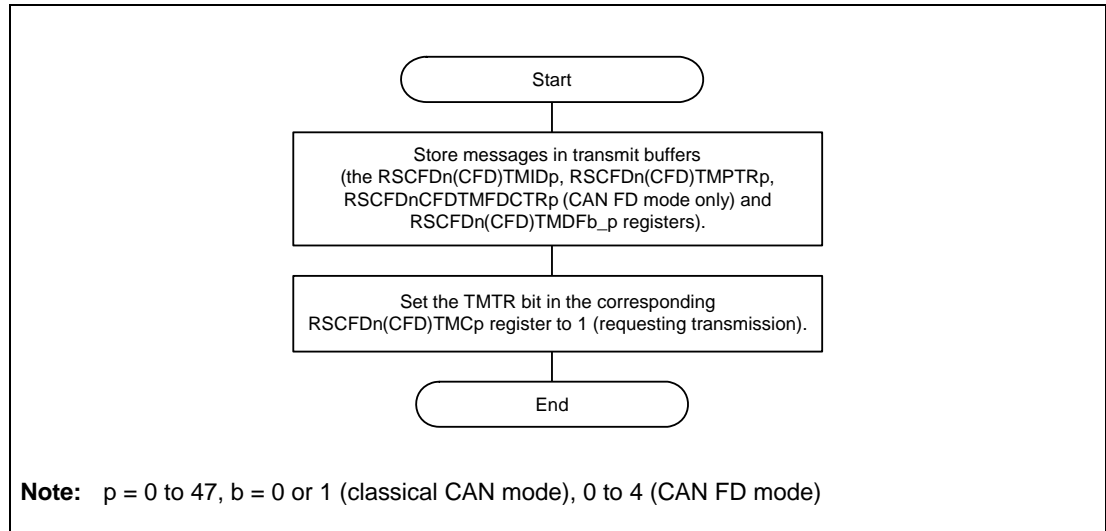


Figure 17.27 Procedure for Transmission from Transmit Buffers

In CAN FD mode and transmit buffer merge mode, messages with a payload size of more than 20 bytes can be transmitted from transmit buffers $(16 \times m) + 0$ and transmit buffers $(16 \times m) + 3$. At this time, transmit buffers $(16 \times m) + 1$ to $(16 \times m) + 2$ and transmit buffers $(16 \times m) + 4$ to $(16 \times m) + 5$ are allocated as a payload storage area. Registers RSCFDnCFDTMIDp, RSCFDnCFDTMPTRp, and RSCFDnCFDTMFDCTRp corresponding to these buffers can be used as data field registers that can store 4-byte data bytes (payload) like the RSCFDnCFDTMDFb_p register. **Table 17.188** shows message storage registers when transmitting a message with a payload size of more than 20 bytes from transmit buffer 0.

Table 17.188 Message Storage Registers in Transmit Buffer Merge Mode (Example of Transmit Buffer 0)

| Transmit Buffer | Offset from Base Address | Symbol | Register Function in Transmit Buffer Merge Mode |
|------------------------|--|--------------------------------------|--|
| Transmit buffer 0 | 4000 _H | RSCFDnCFDTMID0 | Transmit buffer 0 ID data, transmit history data store enable bit, RTR bit, and IDE bit |
| | 4004 _H | RSCFDnCFDTMPTR0 | Transmit buffer 0 label data and DLC data |
| | 4008 _H | RSCFDnCFDTMFDCTR0 | Transmit buffer 0 ESI bit, BRS bit, and FDF bit |
| | 400C _H to 401C _H | RSCFDnCFDTMDF0_0 to RSCFDnCFDTMDF4_0 | Transmit buffer 0 data bytes 0, 1, 2, and 3 to transmit buffer 0 data bytes 16, 17, 18, and 19 |
| Transmit buffer 1 | 4020 _H | RSCFDnCFDTMID1 | Transmit buffer 0 data bytes 20, 21, 22, and 23 |
| | 4024 _H | RSCFDnCFDTMPTR1 | Transmit buffer 0 data bytes 24, 25, 26, and 27 |
| | 4028 _H | RSCFDnCFDTMFDCTR1 | Transmit buffer 0 data bytes 28, 29, 30, and 31 |
| | 402C _H to 403C _H | RSCFDnCFDTMDF0_1 to RSCFDnCFDTMDF4_1 | Transmit buffer 0 data bytes 32, 33, 34, and 35 to transmit buffer 0 data bytes 48, 49, 50, and 51 |
| Transmit buffer 2 | 4040 _H | RSCFDnCFDTMID2 | Transmit buffer 0 data bytes 52, 53, 54, and 55 |
| | 4044 _H | RSCFDnCFDTMPTR2 | Transmit buffer 0 data bytes 56, 57, 58, and 59 |
| | 4048 _H | RSCFDnCFDTMFDCTR2 | Transmit buffer 0 data bytes 60, 61, 62, and 63 |
| | 404C _H to 405C _H | RSCFDnCFDTMDF0_2 to RSCFDnCFDTMDF4_2 | Not used |

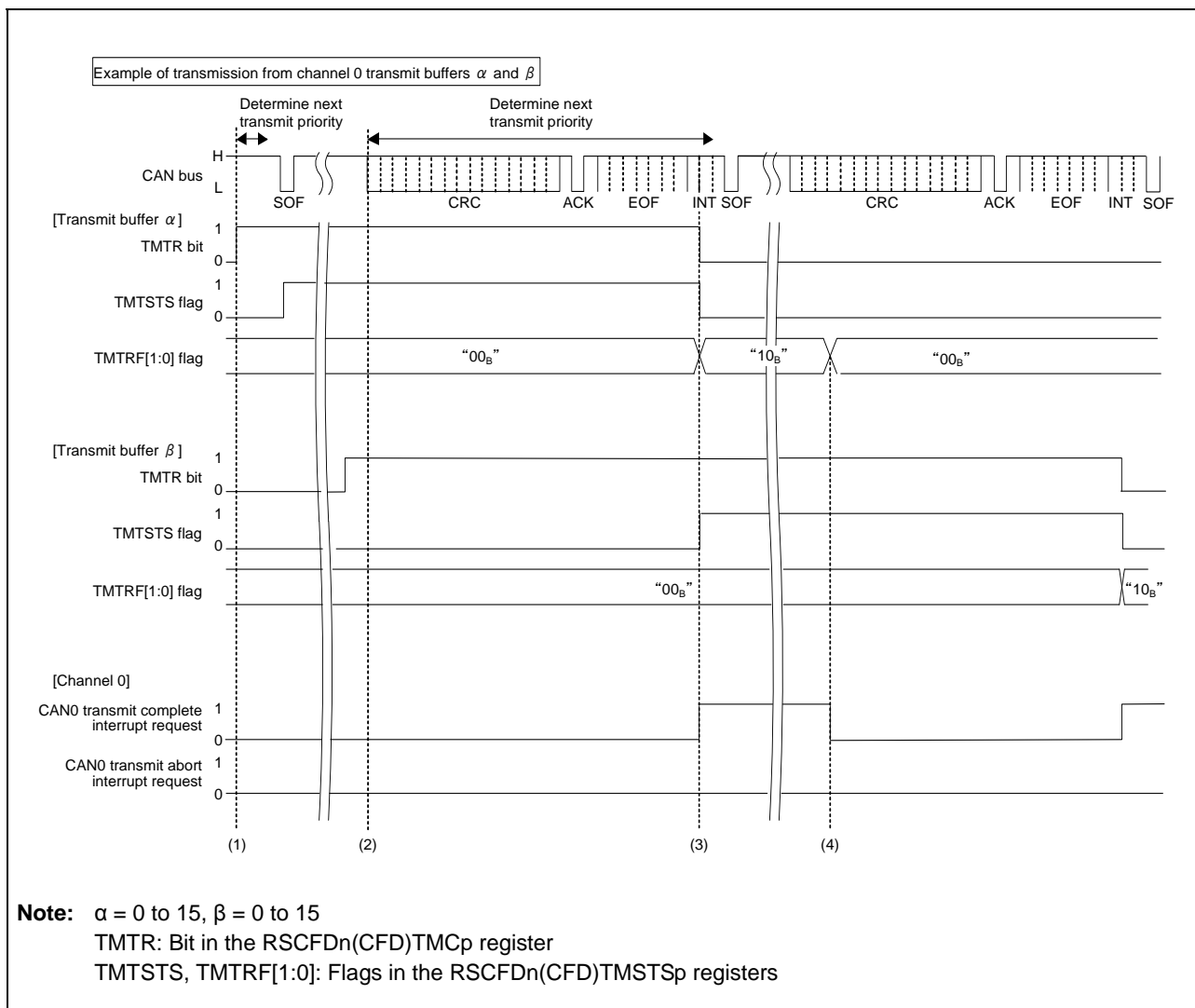


Figure 17.28 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

- (1) When the TMTR bit in the RSCFDn(CFD)TMC α register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer α is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCFDn(CFD)TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.
- (2) When a transmit request from a buffer is present, the priority determination starts at the first bit of CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur between transmissions because the determination processing is completed by the third bit of the intermission.
- (3) When transmission completes successfully, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSa register is set to 10_B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCFDn(CFD)TMC α register are cleared to 0. When the TMIE α bit in the RSCFDn(CFD)TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

- (4) Before starting the next transmission, set the TMTRF[1:0] flag to 00_B. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is 00_B.

If an arbitration-lost occurs after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted (in classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1).

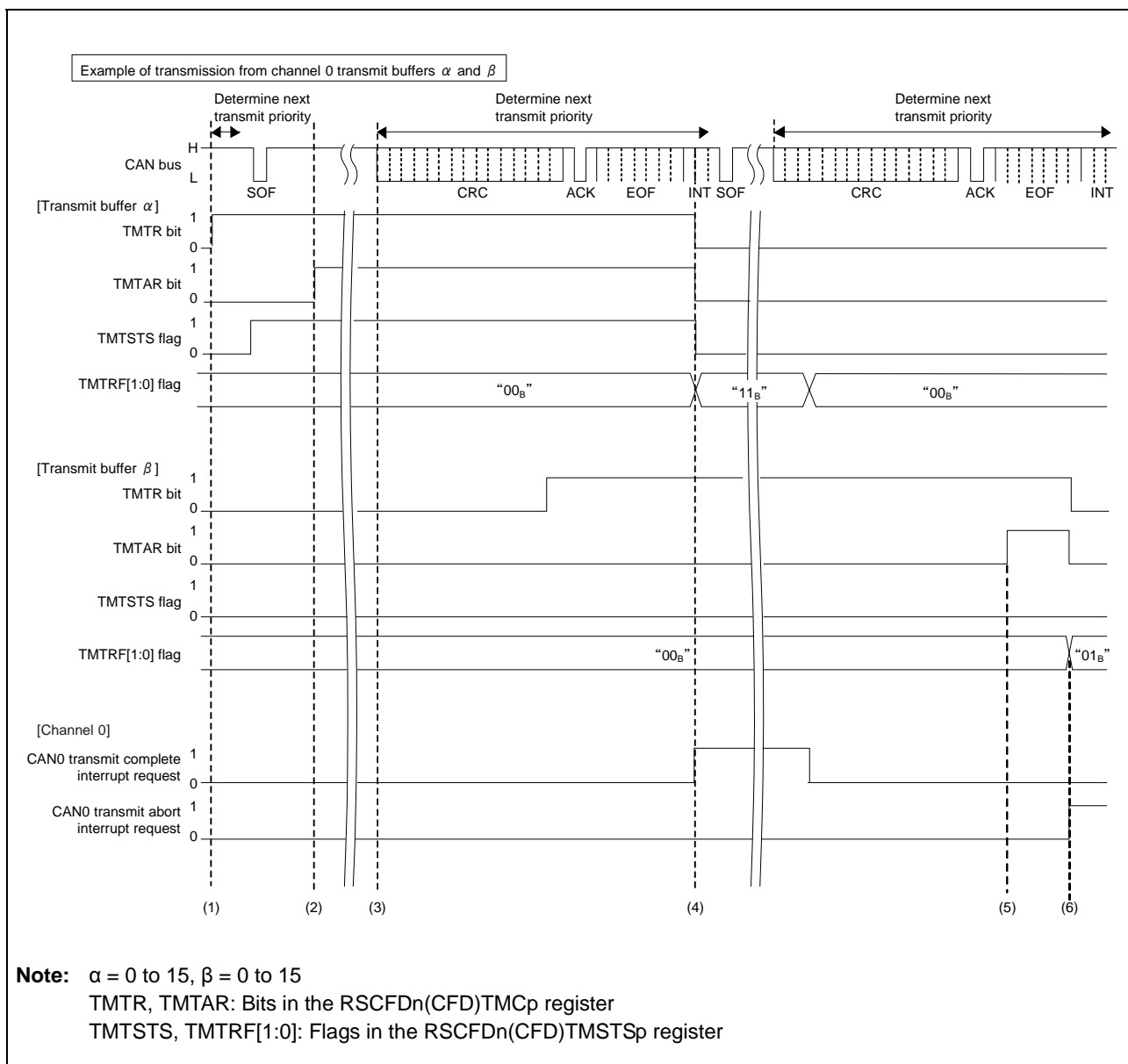


Figure 17.29 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

- (1) When the TMTR bit in the RSCFDn(CFD)TMC α register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer α is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCFDn(CFD)TMSTS α register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.
- (2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration loss occurs even if the TMTAR bit is set to 1 (transmit abort is requested).
- (3) The priority determination starts at the first bit of the CRC field for the next transmission. In this timing chart, buffer β is not selected as the next transmit buffer. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.
- (4) When transmission completes successfully, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTS α register is set to 11_B (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCFDn(CFD)TMC α register are cleared to 0. When the TMIE α value in the RSCFDn(CFD)TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).
- (5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.
- (6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01_B. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01_B. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RSCFDn(CFD)CmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B.

If an arbitration loss occurs after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted (in classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1).

17.11.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 17.30 shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 17.31 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. Figure 17.32 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.

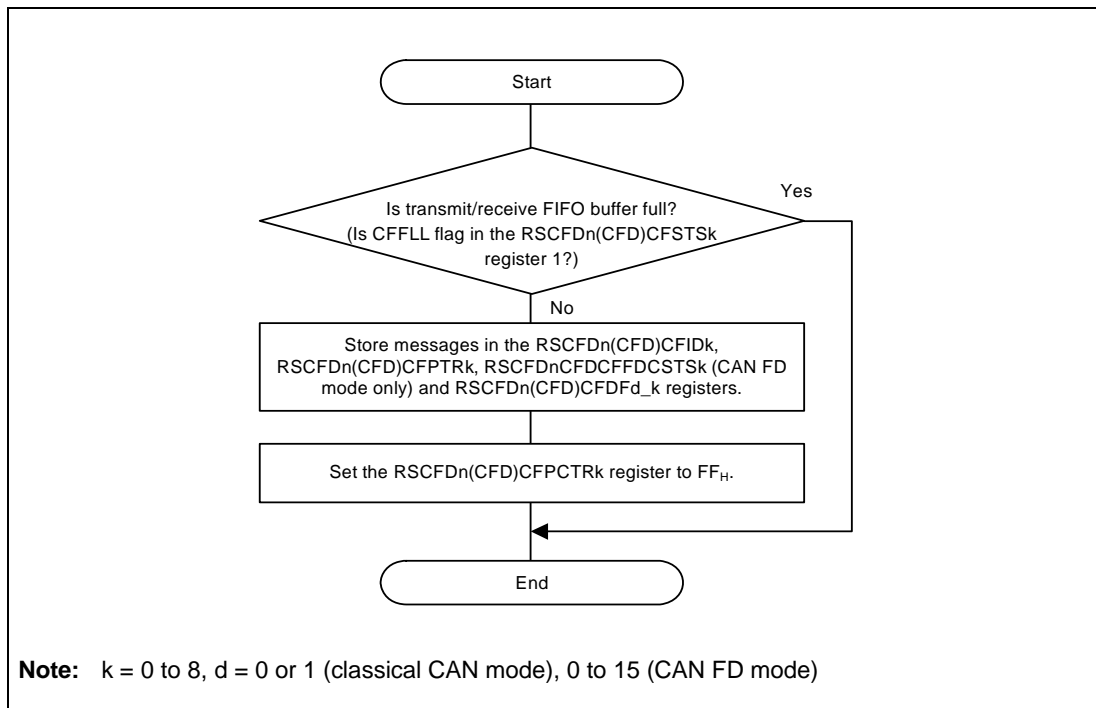


Figure 17.30 Procedure for Transmission from Transmit/Receive FIFO Buffers

When storing a message, do not write a value to the RSCFDnCFDCDFDk register corresponding to the area exceeding the payload storage size specified by the CFPLS[2:0] bits in the RSCFDnCFDCFCCK register.

Table 17.189 Payload Storage Area of Transmit/Receive FIFO Buffer

| Set CFPLS[2:0] Value | Payload Storage Size | Corresponding Data Field Registers |
|----------------------|----------------------|---------------------------------------|
| 000 _B | 8 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD1_k |
| 001 _B | 12 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD2_k |
| 010 _B | 16 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD3_k |
| 011 _B | 20 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD4_k |
| 100 _B | 24 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD5_k |
| 101 _B | 32 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD7_k |
| 110 _B | 48 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD11_k |
| 111 _B | 64 bytes | RSCFDnCFDCDFD0_k to RSCFDnCFDCDFD15_k |

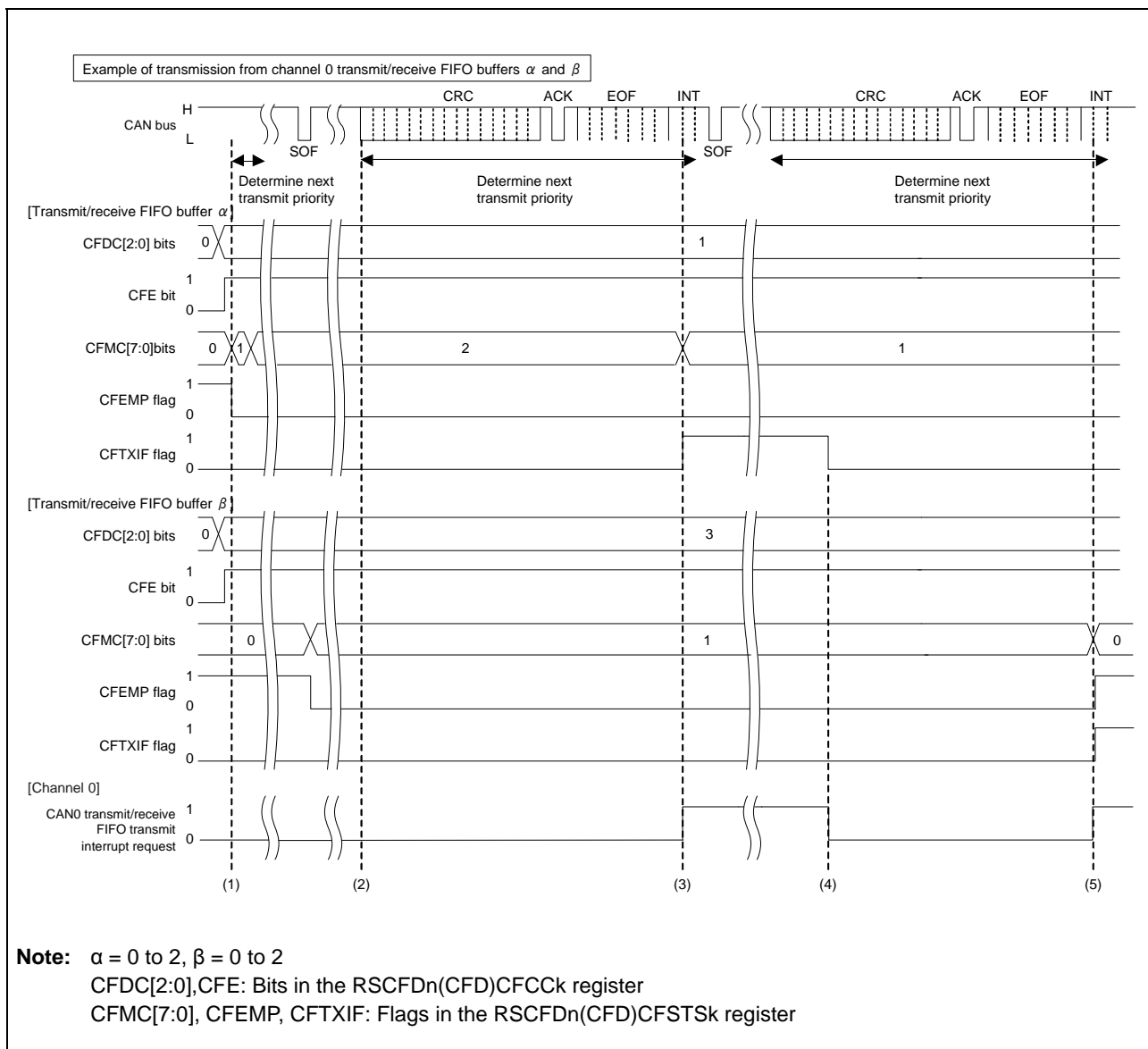


Figure 17.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

- (1) While the CAN bus is idle, when the CFE bit in the RSCFDn(CFD)CFCC α register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCFDn(CFD)CFCC α register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer α of channel 0.
- (2) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

- (3) When transmission completes successfully, the CFMC[7:0] value in the RSCFDn(CFD)CFSTS α register is decremented by 1. Setting the CFIM bit in the RSCFDn(CFD)CFCC α register to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCFDn(CFD)CFSTS α register to 1 (a transmit/receive FIFO transmit interrupt request is present).
- (4) The program can clear the CFTXIF flag.
- (5) Message transmission from transmit/receive FIFO buffer β of channel 0 completes and the CFMC[7:0] value in the RSCFDn(CFD)CFSTS β register is decremented by 1. The CFMC[7:0] bits are cleared to 00_H and therefore the CFEMP flag in the RSCFDn(CFD)CFSTS β register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RSCFDn(CFD)CFSTS α and RSCFDn(CFD)CFSTS β registers is set to 1 (the transmit/receive FIFO buffer is full).

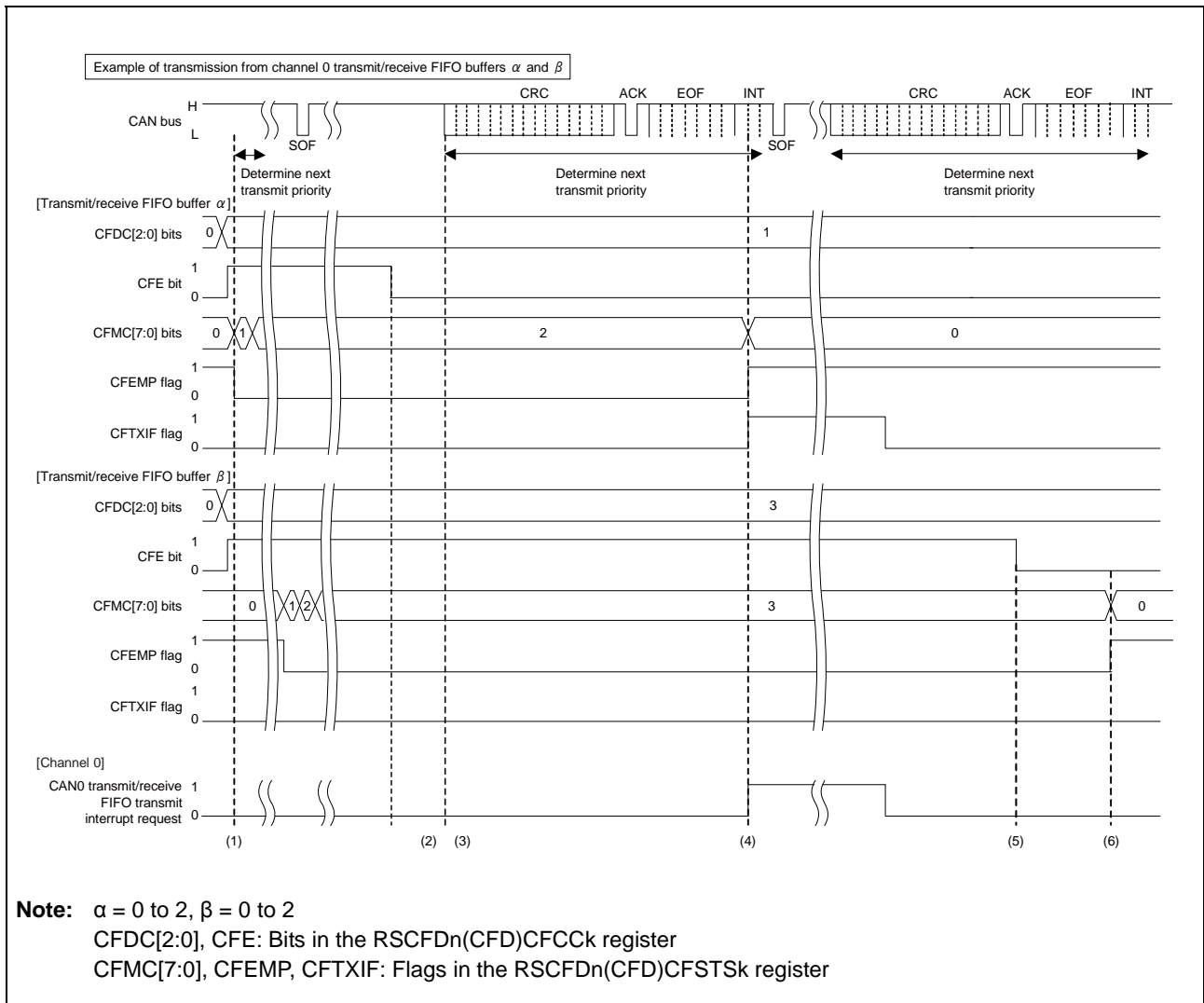


Figure 17.32 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

- (1) While the CAN bus is idle, when the CFE bit in the RSCFDn(CFD)CFCC α register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCFDn(CFD)CFCC α register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCFDn(CFD)CFSTS α register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer α of channel 0.
- (2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration loss occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).
- (3) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. In this figure, transmit/receive FIFO buffer β is not selected as a buffer for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.
- (4) When transmit completes successfully, the CFMC[7:0] value is cleared to 00_H. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCFDn(CFD)CFSTS α register to 1 (a transmit/receive FIFO transmit interrupt request is present). The program can clear the CFTXIF flag.
- (5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer β), transmit/receive FIFO buffer β cannot be disabled immediately even if the CFE bit in the RSCFDn(CFD)CFCC β register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RSCFDn(CFD)CFSTS β register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)
- (6) After the internal processing time has passed, transmit/receive FIFO buffer β are disabled and the CFMC[7:0] bits in the RSCFDn(CFD)CFSTS β register are cleared to 00_H and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer β is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer β is immediately disabled. (The CFMC[7:0] bits are cleared to 00_H and the CFEMP flag is set to 1.)

17.11.3.3 Procedure for Transmission from the Transmit Queue

Figure 17.33 shows the procedure for transmission from the transmit queue.

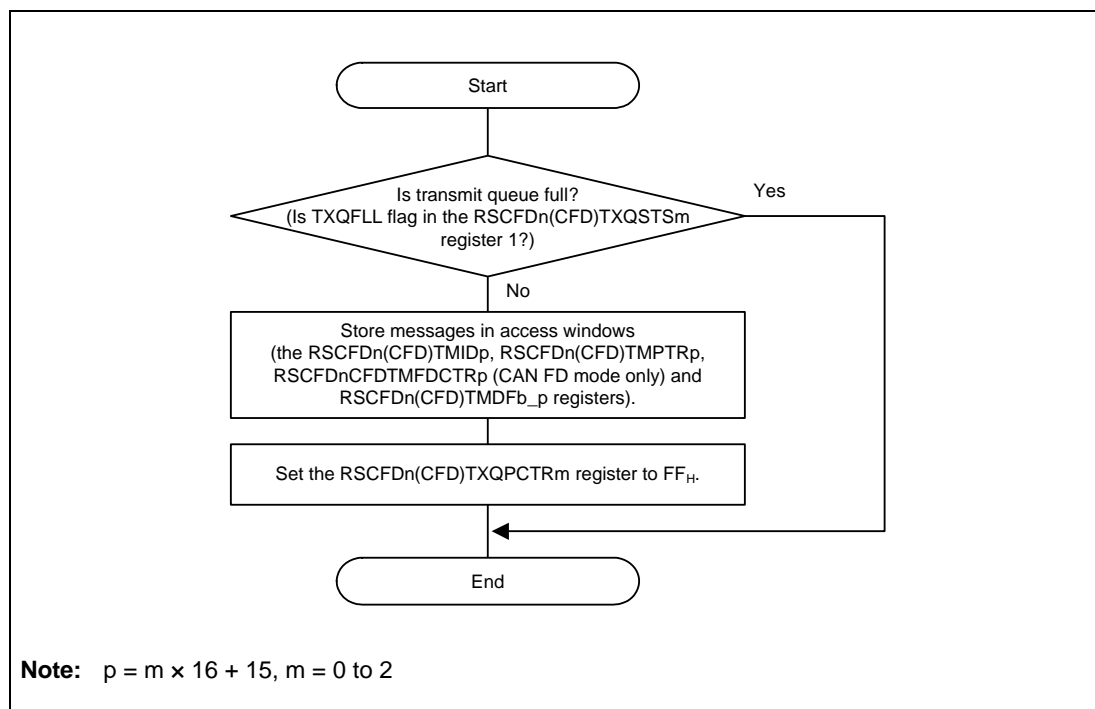


Figure 17.33 Procedure for Transmission from the Transmit Queue

17.11.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RSCFDn(CFD)THLACCm register. The next data can be accessed by writing FF_H to the corresponding RSCFDn(CFD)THLPCTRm register (m = 0 to 2) after reading a set of data. **Figure 17.34** shows the transmit history buffer reading procedure.

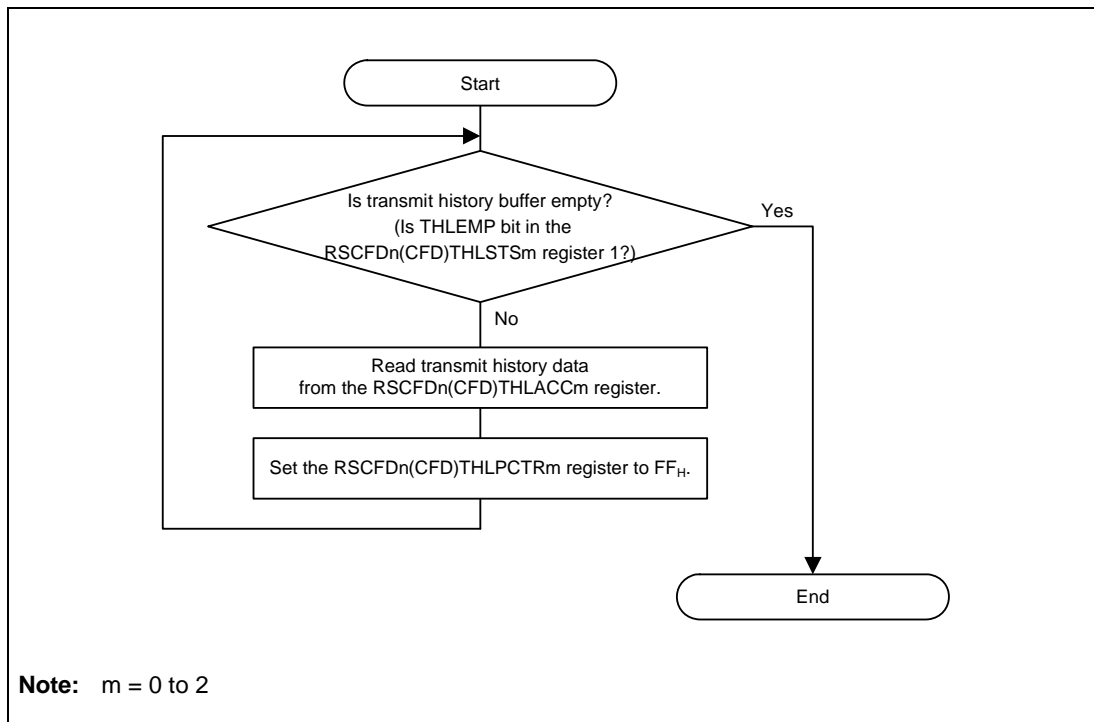


Figure 17.34 Transmit History Buffer Reading Procedure

17.11.4 Test Settings

17.11.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by enabling a CAN node to receive its own transmitted messages.

Figure 17.35 shows the self-test mode setting procedure.

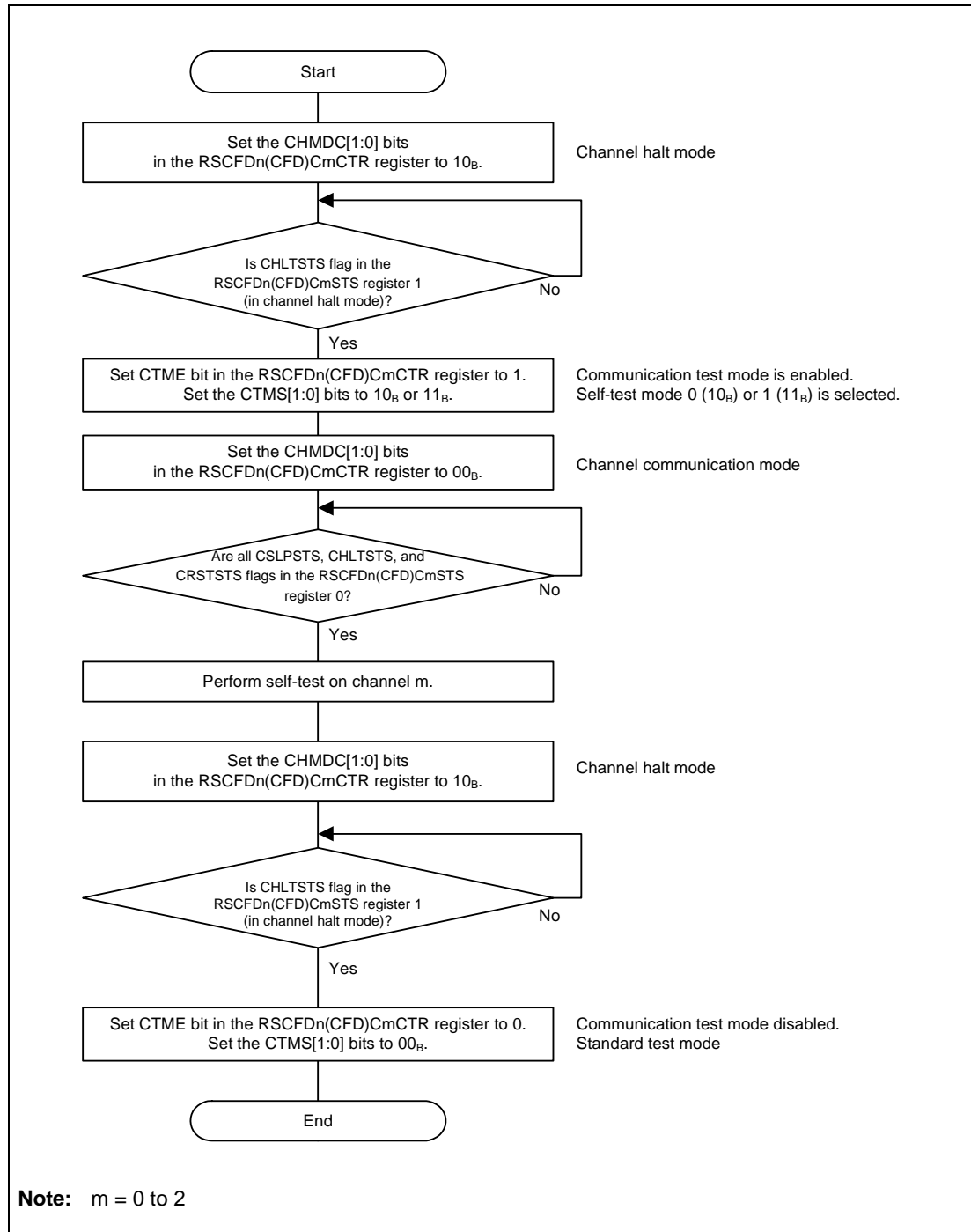


Figure 17.35 Self-Test Mode Setting Procedure

17.11.4.2 Procedure for Releasing the Protection

Since the global test function in **Table 17.190** is protected, write the protection release data 1 and release data 2 in succession to the LOCK[15:0] bits in the RSCFDn(CFD)GLOCKK register, then set the target test bit to 1.

Table 17.190 Protection Release Data for Test Function

| Test Function | Protection Release Data 1 | Protection Release Data 2 | Target Bit |
|---------------|---------------------------|---------------------------|---|
| RAM test | 7575 _H | 8A8A _H | RTME bit in the RSCFDn(CFD)GTSTCTR register |

If an incorrect value is written to the LOCK[15:0] bits, restart from writing the protection release data 1. **Figure 17.36** shows the procedure for releasing the protection.

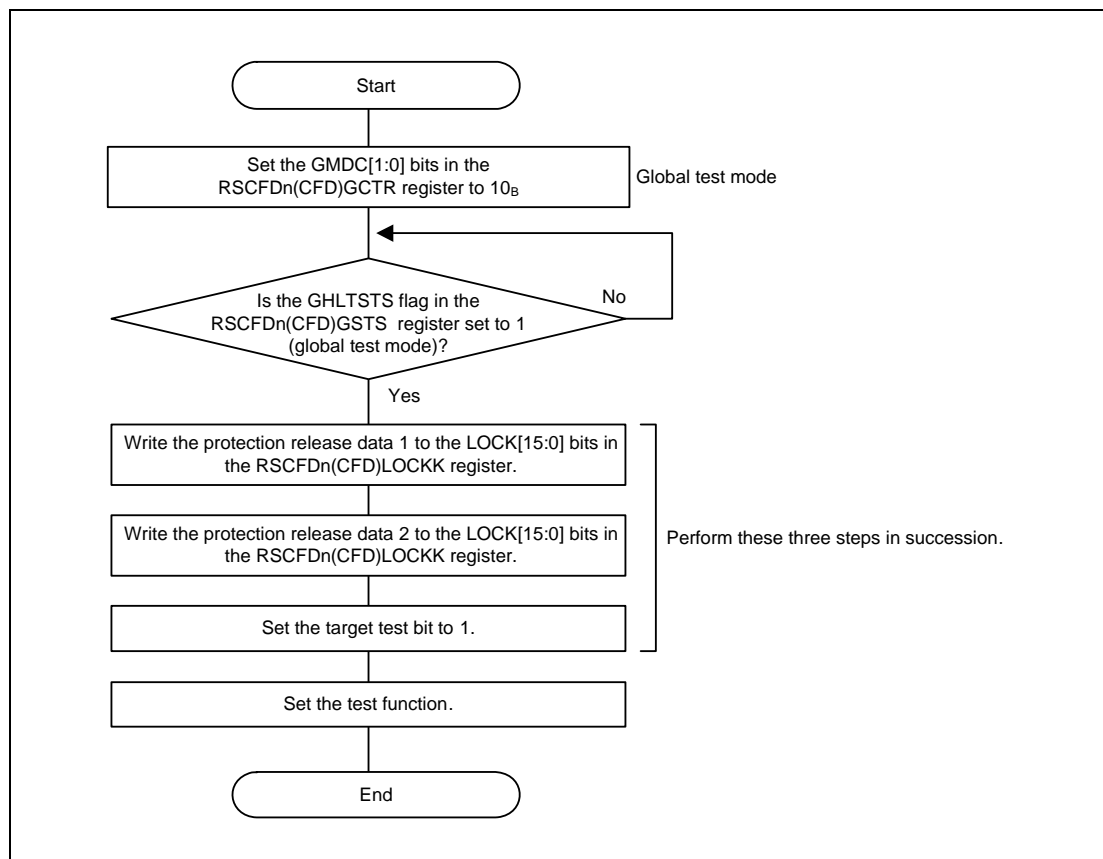


Figure 17.36 Protection Release Procedure

17.11.4.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before ending the RAM test, write 0000 0000_H to all pages of the CAN RAM.

Figure 17.37 shows the RAM test setting procedure.

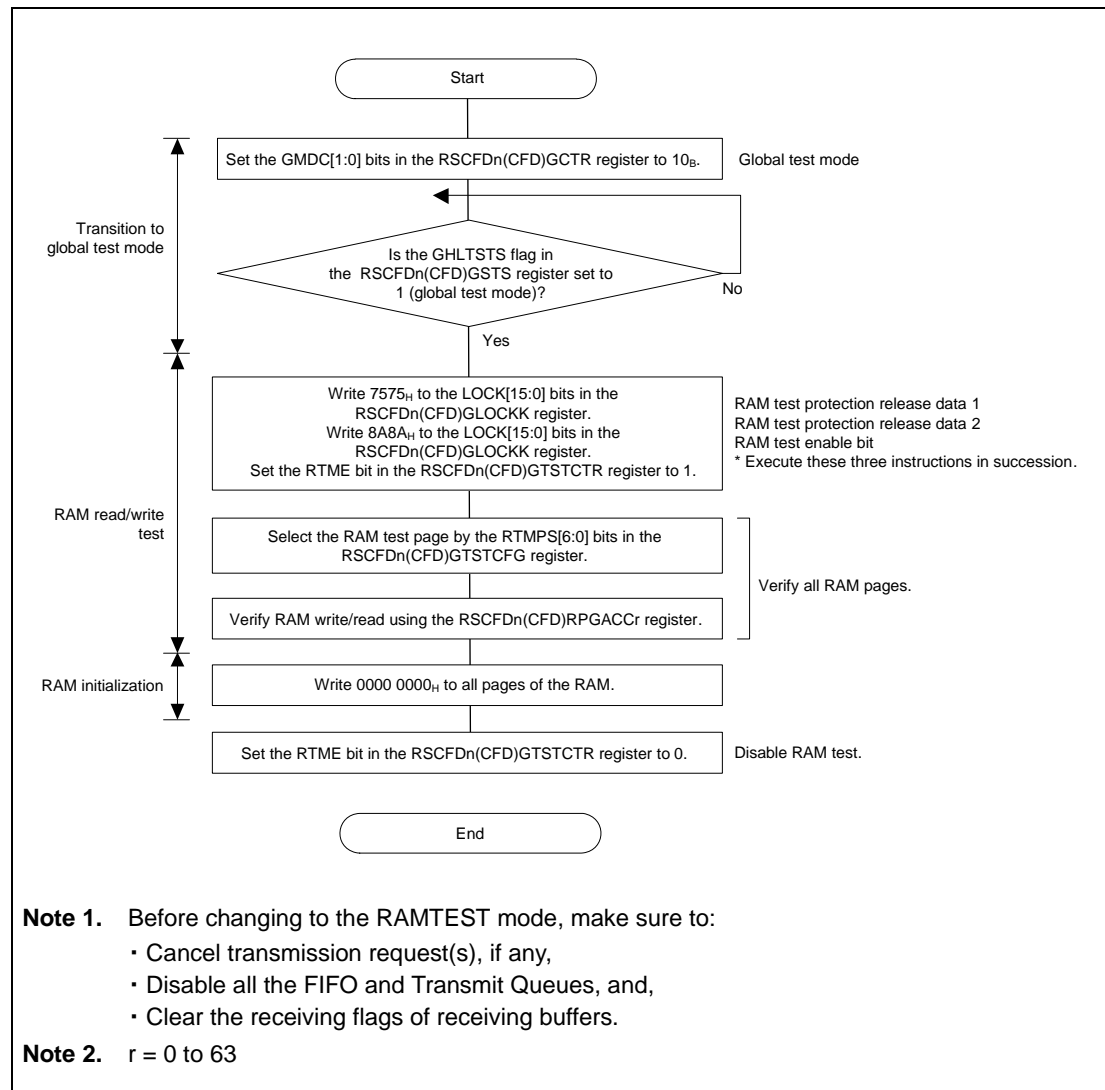


Figure 17.37 RAM Test Setting Procedure

17.11.4.4 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

Figure 17.38 shows the inter-channel communication test setting procedure.

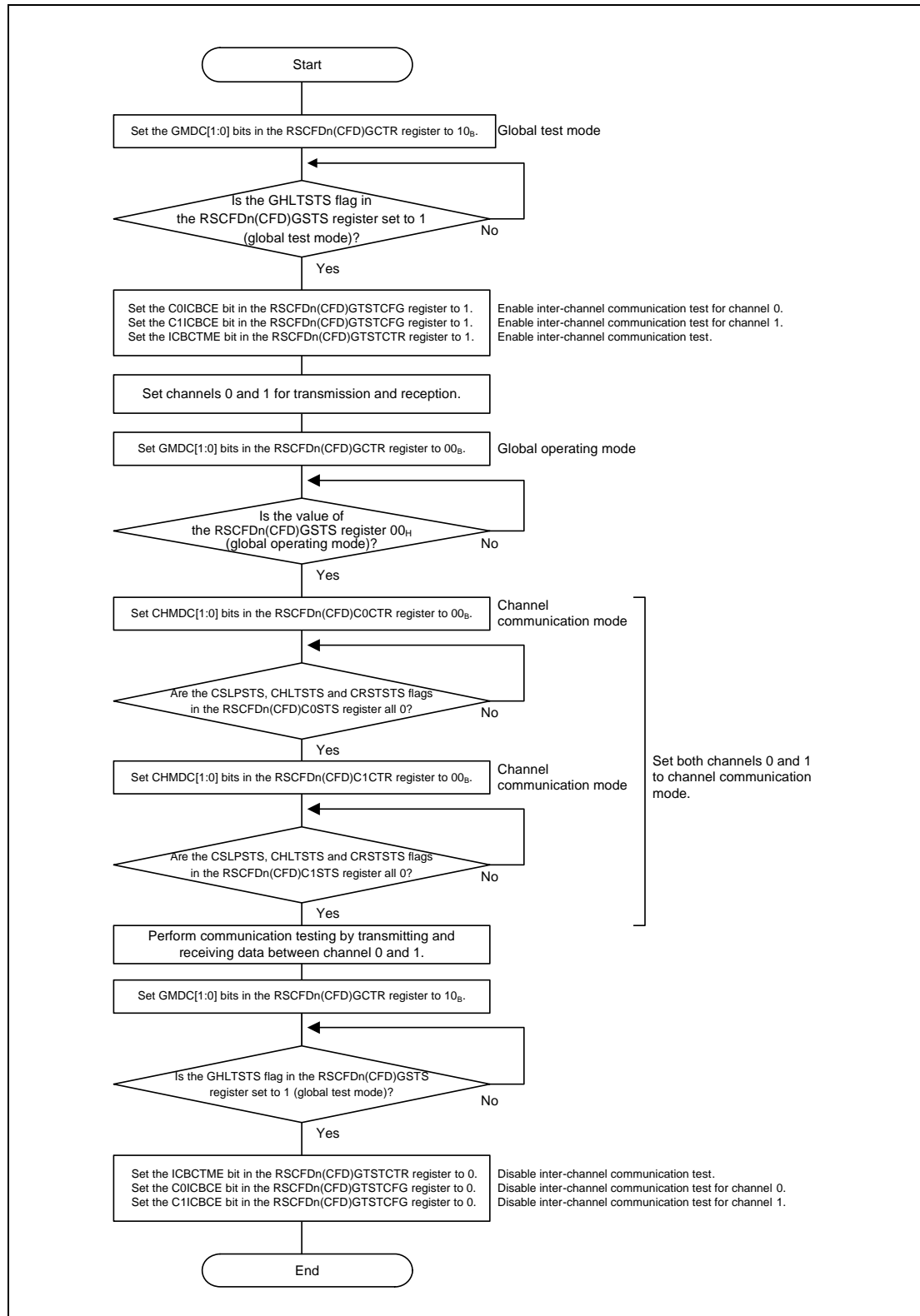


Figure 17.38 Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

17.12 Notes on the RS-CANFD Module

- When changing interface mode without resetting the RS-CANFD, write the value after reset to all registers and bits that are not allocated to the register map after change and then modify the RSCFDn(CFD)GRMCFG register.
- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RSCFDn(CFD)GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RSCFDn(CFD)CmSTS register (m = 0 to 2) for transitions.
- When only classical CAN frames are used in CAN FD mode, set the RSCFDnCFDCmDCFG register to the value equal to the set RSCFDnCFDCmNCFG register value.
- The acceptance filter processing checks receive rules sequentially in ascending order from the smallest rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the smallest number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RSCFDn(CFD)TMCp) of the corresponding transmit buffer to 00_H. The status register (RSCFDn(CFD)TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers RSCFDn(CFD)TMTRSTSy, RSCFDn(CFD)TMTARSTSy, RSCFDn(CFD)TMTCASTSy, and RSCFDn(CFD)TMTASTSy), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues remain unchanged. Set the enable bit in the corresponding interrupt enable register (registers RSCFDn(CFD)TMIECy) to 0 (transmit buffer interrupt is disabled).
- When using transmit buffer merge mode (in CAN FD mode), write 00_H to the control register (RSCFDn(CFD)TMCp) of the transmit buffer corresponding to the transmit buffer allocated as a payload storage area. Set the enable bit of corresponding interrupt enable registers (RSCFDn(CFD)TMIECy) to 0 (to disable interrupts).
- Transmit buffers that are linked to transmit/receive FIFO buffers must not be allocated to transmit queues. Do not allocate a transmit buffer allocated as a payload storage area in transmit buffer merge mode (in CAN FD mode) to the transmit queue either.
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to transmit buffers of the same number.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a newly received message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.

- In the case of registers that access the RAM, the value after reset shown in **Section 17.3, Registers (Classical CAN Mode)** and **Section 17.4, Registers (CAN FD Mode)** indicate the values cleared by initialization of the CAN RAM. Values before clear are undefined. The following registers apply.
 - Receive rule (RSCFDn(CFD)GAFLIDj, RSCFDn(CFD)GAFLMj, RSCFDn(CFD)GAFLP0_j, RSCFDn(CFD)GAFLP1_j registers)
 - Receive buffers (RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDn(CFD)CFDRMFDSTSq, RSCFDn(CFD)RMDfB_q registers)
 - Receive FIFO buffer access registers (RSCFDn(CFD)RFIDx, RSCFDn(CFD)RFPTRx, RSCFDn(CFD)CFDRFFDSTSx, and RSCFDn(CFD)RFDFd_x registers)
 - Transmit/receive FIFO buffer access registers (RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDn(CFD)CFDCFFDCSTSk, and RSCFDn(CFD)CFDFd_k registers)
 - Transmit buffers (RSCFDn(CFD)TMIDp, RSCFDn(CFD)TMPTRp, RSCFDn(CFD)CFDTMFDCTRp, and RSCFDn(CFD)TMDFb_p registers)
 - Transmit history access register (RSCFDn(CFD)THLACCm register)
 - RAM test page access register (RSCFDn(CFD)RPGACCr register)
- The values of unused receive buffers (RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDn(CFD)CFDRMFDSTSq, and RSCFDn(CFD)RMDfB_q registers), receive FIFO buffer access registers (RSCFDn(CFD)RFIDx, RSCFDn(CFD)RFPTRx, RSCFDn(CFD)CFDRFFDSTSx, and RSCFDn(CFD)RFDFd_x registers) and transmit/receive FIFO buffer access registers (RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDn(CFD)CFDCFFDCSTSk, and RSCFDn(CFD)CFDFd_k registers) are undefined when the RS-CANFD module transitions to global operation mode or global test mode after exiting from global reset mode.

Section 18 FlexRay (FLXA)

The FlexRay IP-module performs communication according to the FlexRay protocol specification v2.1. With maximum specified sample clock the bitrate is 10 MBit/s. Additional bus driver (BD) hardware is required for connection to the physical layer.

18.1 Features of RH850/P1M-E FLXA

18.1.1 Number of Units and Channels

This microcontroller has the following number units of FlexRay.

Table 18.1 Number of Units

| FlexRay | RH850/P1M-E |
|-----------------|----------------|
| Number of Units | 1 (A ch, B ch) |
| Name | FLXAn |

Table 18.2 Index

| Index | Meaning |
|-------|--|
| n | The number of the FlexRay indicated by the letter "n" (n = 0). |
| m | The register number is identified by the index "m", for example, FLXAnFRESIDm for the even Sync ID register. |
| p | The flag number is indicated by the letter "p" ($p = (m - 1) \times 32$ to $(m \times 32 - 1)$). |

18.1.2 Register Base Address

All FlexRay register addresses are given as address offsets to the individual base address <FLXAn_base>.

The base address <FLXAn_base> of each FlexRay is listed in the following table:

Table 18.3 Register Base Address <FLXAn_base>

| Base Address Name | Base Address |
|-------------------|------------------------|
| <FLXA0_base> | 1002 0000 _H |

18.1.3 Clock Supply

FlexRay provide one clock input.

Table 18.4 Clock Supply

| Unit Name | Clock for the Unit | Supply Clock Name |
|-----------|--------------------------------------|-------------------------------------|
| FLXAn | Sample clock Peripheral bus clock | High-speed peripheral clock CLK_HSB |

18.1.4 Interrupt Requests

FlexRay can generate the following interrupt requests:

Table 18.5 Interrupt Requests

| Unit Interrupt Name | Outline | Interrupt | DMA/DTS Trigger |
|---------------------|---|-----------|-----------------|
| INTFLXA0LINE0 | FLXA0 FlexRay0 interrupt | 194 | — |
| INTFLXA0LINE1 | FLXA0 FlexRay1 interrupt | 195 | — |
| INTFLXA0TIM0 | FLXA0 Timer 0 interrupt | 196 | — |
| INTFLXA0TIM1 | FLXA0 Timer 1 interrupt | 197 | — |
| INTFLXA0TIM2 | FLXA0 Timer 2 interrupt | 198 | — |
| INTFLXA0FDA | FLXA0 FIFO transfer interrupt | 199 | — |
| INTFLXA0FW | FLXA0 FIFO transfer warning interrupt | 200 | — |
| INTFLXA0OW | FLXA0 Output transfer warning interrupt | 201 | — |
| INTFLXA0OT | FLXA0 Output transfer end interrupt | 202 | — |
| INTFLXA0IQF | FLXA0 Input queue full interrupt | 203 | — |
| INTFLXA0IQE | FLXA0 Input queue empty interrupt | 204 | — |

18.1.5 Reset Source

Please refer to **Section 8, Reset Controller**.

18.1.6 External Input/Output Pins

Table 18.6 shows the pin information.

Table 18.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signals |
|------------------|-----|--------------------------|------------------------------|
| rxda_extfxr | I | Receive Data(ch:A) | FLXA0RXDA |
| rxdb_extfxr | I | Receive Data(ch:B) | FLXA0RXDB |
| fxr_txda | O | Transmit Data(ch:A) | FLXA0TXDA |
| fxr_txdb | O | Transmit Data(ch:B) | FLXA0TXDB |
| fxr_txena_n | O | Transmit Enable(ch:A) | FLXA0TXENA |
| fxr_txenb_n | O | Transmit Enable(ch:B) | FLXA0TXENB |
| stpwt_extfxr | I | Stop watch trigger input | FLXA0STPWT |

18.1.7 Combinations of Pins and Ports

Combinations of FlexRay pins and ports are listed in the following table.

Table 18.7 Combinations of Pins and Ports

| Function | Pin Name | Port Name | |
|----------|------------|-----------|---------------------|
| | | Group 1 | Group 2 |
| FLXA0 | FLXA0RXDA | P4_2 | P4_8* ¹ |
| | FLXA0RXDB | P4_3 | P4_11* ¹ |
| | FLXA0STPWT | P4_4 | P4_14* ¹ |
| | FLXA0TXDA | P4_0 | P4_9* ¹ |
| | FLXA0TXDB | P4_5 | P4_12* ¹ |
| | FLXA0TXENA | P4_1 | P4_10* ¹ |
| | FLXA0TXENB | P4_6 | P4_13* ¹ |

Note 1. Available in devices with 144-pin.

18.1.8 Functions

For communication on a FlexRay network, individual message buffers with up to 254 data bytes are configurable. The message buffer is a Message RAM that is configurable up to 128 message buffers. All functions concerning the handling of messages are implemented in the Message Handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the Message RAM, maintaining the transmission schedule as well as providing message status information.

The register set of the FlexRay IP-module can be accessed directly by an external Host via the module's Host interface. These registers are used to control/configure/monitor the FlexRay Channel Protocol Controllers, Message Handler, Global Time Unit, System Universal Control, Frame and Symbol Processing, Network Management, Interrupt Control, to access the Message RAM via Input / Output Buffer; and to control the data transfer between the Message RAM and the Local RAM/Global RAM.

The FlexRay IP-module supports the following features:

| Item | Specification |
|---------------------------------|--|
| Communication | Conformance with FlexRay protocol specification v2.1 |
| Data transfer rate | Up to 10 M bit/s on each channel |
| Data link layer clock frequency | 80 MHz |
| Input/Output pins per channel | TxD, RxD, TxEN |
| FlexRay channels | 2 (channels A and B) |
| Message buffer | Up to 128 message buffers are configurable. Message buffers are configurable with different payload length Each message buffer can be configured as a part of receive buffer, transfer buffer or receive FIFO. Filtering for slot counter, cycle counter or channel. |
| Message RAM | 8-Kbyte message RAM can be configured as below. <ul style="list-style-type: none"> • 128 message buffers with up to 48-byte data section • 30 message buffers with 254-byte data section |
| FIFO | One configurable receive FIFO |
| Message Buffer Access | Access by the host CPU through input/output buffer Input buffer: a message transferred to the message RAM is retained. Output buffer: a message read from the message RAM is retained. Access by data transfer function Input transfer: the contents of the message buffer is transferred from the Local RAM/Global RAM to the message RAM by a request of the CPU Input transfer: the contents of the message buffer is transferred automatically to the Local RAM/Global RAM from the message RAM |
| Network management | Supported |
| Interrupts | Maskable module interrupts |
| Timer | Two absolute timers One relative timer One stop watch timer |

18.1.9 Block Diagram

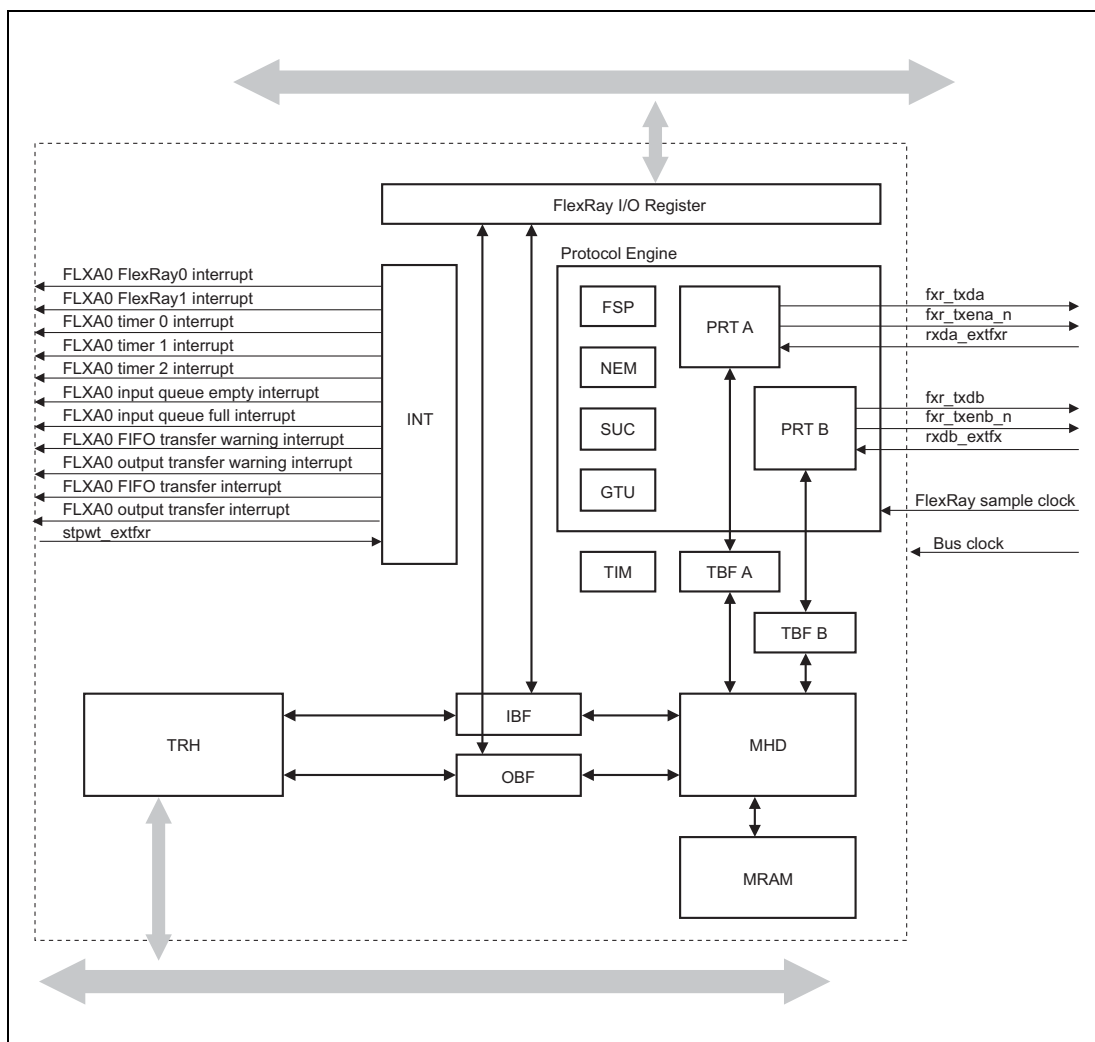


Figure 18.1 FlexRay IP Block Diagram

Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host can read the header and data section of the transferred message buffer from the Output Buffer.

Message Handler (MHD)

The FlexRay Message Handler controls data transfers between the following components:

- Input / Output Buffer and Message RAM
- TBFRAMs of the two FlexRay Protocol Controllers and Message RAM

Message RAM (MRAM)

The Message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

TBFRAM (TBF A/B)

Stores the data section of two complete messages.

FlexRay Channel Protocol Controller (PRT A/B)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the TBFRAMs for intermediate message storage and to the physical layer via bus driver (BD).

They perform the following functionality:

- Control and check of bit timing
- Reception / transmission of FlexRay frames and symbols
- Check of header CRC
- Generation / check of frame CRC
- Interfacing to bus driver

Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick (μT)
- Generation of macrotick (MT)
- Fault tolerant clock synchronization by FTM algorithm
 - rate correction
 - offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

System Universal Control (SUC)

The System Universal Control controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation

Frame and Symbol Processing (FSP)

The Frame and Symbol Processing controls the following functions:

- Checks the correct timing of frames and symbols
- Tests the syntactical and semantical correctness of received frames
- Sets the slot status flags

Network Management (NEM)

Handles the network management vector.

Interrupt Control (INT)

The Interrupt Controller performs the following functions:

- Provides error and status interrupt flags
- Enable / disable interrupt sources
- Assignment of interrupt sources to one of the two general module interrupt lines
- Enable / disable module interrupt lines

Timer (TIM)

The Timer module includes the following macrotick timer:

- two absolute timers
- one relative timer
- one stop watch timer

Transfer Handler (TRH)

Handles the data transfer between Local RAM/Global RAM and FlexRay module.

The Transfer Handler supports the following transfer types:

- Transfer of buffer configuration data from the Local RAM/Global RAM to the Message RAM
- Transfer of payload data for transmission buffers from the Local RAM/Global RAM to the Message RAM
- Transfer of buffer configuration data and payload data for transmission buffer from the Local RAM/Global RAM to the Message RAM
- Automatic transfer of payload data from receive buffer to the Local RAM/Global RAM upon frame reception
- Automatic transfer of payload data, buffer configuration data and message buffer status data from receive buffer to the Local RAM/Global RAM upon frame reception
- Automatic transfer of buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Global RAM in response to slot status update
- Manual transfer of payload data, buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Global RAM.

18.2 Register

18.2.1 Register Map

The FlexRay module allocates an address space as shown in **Table 18.8**.

Within this specification the “Values after reset” refers to the reset of the microcontroller. For registers in the address range $\langle \text{FLXAn_base} \rangle + 0010_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 0\text{FFF}_{\text{H}}$ the “Value after reset” are also applicable when the software reset of the FlexRay module (using bit FLXAnFROC.OE) is applied.

The addresses in this specification are listed as offsets from a base address. The base address $\langle \text{FLXAn_base} \rangle$ must thus be added to the addresses.

For $\langle \text{FLXAn_base} \rangle$, refer to **Section 18.1.2, Register Base Address**.

Table 18.8 FlexRay Register Map (1/3)

| Register Name | Symbol | Value after Reset | Address | Access Size |
|--|--------------|------------------------|--|-------------|
| FlexRay Operation Control Register | FLXAnFROC | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0004_{\text{H}}$ | 8, 16, 32 |
| FlexRay Operation Status Register | FLXAnFROS | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 000\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay Lock Register | FLXAnFRLCK | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 001\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay Error Interrupt Register | FLXAnFREIR | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0020_{\text{H}}$ | 8, 16, 32 |
| FlexRay Status Interrupt Register | FLXAnFRSIR | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0024_{\text{H}}$ | 8, 16, 32 |
| FlexRay Error Interrupt Line Select | FLXAnFREILS | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0028_{\text{H}}$ | 8, 16, 32 |
| FlexRay Status Interrupt Line Select | FLXAnFRSILS | 0303 FFFF _H | $\langle \text{FLXAn_base} \rangle + 002\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay Error Interrupt Enable Set Register | FLXAnFREIES | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0030_{\text{H}}$ | 8, 16, 32 |
| FlexRay Error Interrupt Enable Reset Register | FLXAnFREIER | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0034_{\text{H}}$ | 8, 16, 32 |
| FlexRay Status Interrupt Enable Set Register | FLXAnFRSIES | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0038_{\text{H}}$ | 8, 16, 32 |
| FlexRay Status Interrupt Enable Reset Register | FLXAnFRSIER | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 003\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay Interrupt Line Enable Register | FLXAnFRILE | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0040_{\text{H}}$ | 8, 16, 32 |
| FlexRay Timer 0 Configuration Register | FLXAnFRT0C | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0044_{\text{H}}$ | 8, 16, 32 |
| FlexRay Timer 1 Configuration Register | FLXAnFRT1C | 0002 0000 _H | $\langle \text{FLXAn_base} \rangle + 0048_{\text{H}}$ | 8, 16, 32 |
| FlexRay Stop Watch Register 1 | FLXAnFRSTPW1 | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 004\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay Stop Watch Register 2 | FLXAnFRSTPW2 | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0050_{\text{H}}$ | 8, 16, 32 |
| FlexRay SUC Configuration Register 1 | FLXAnFRSUCC1 | 0C40 1080 _H | $\langle \text{FLXAn_base} \rangle + 0080_{\text{H}}$ | 8, 16, 32 |
| FlexRay SUC Configuration Register 2 | FLXAnFRSUCC2 | 0100 0504 _H | $\langle \text{FLXAn_base} \rangle + 0084_{\text{H}}$ | 8, 16, 32 |
| FlexRay SUC Configuration Register 3 | FLXAnFRSUCC3 | 0000 0011 _H | $\langle \text{FLXAn_base} \rangle + 0088_{\text{H}}$ | 8, 16, 32 |
| FlexRay NEM Configuration Register | FLXAnFRNEMC | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 008\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay PRT Configuration Register 1 | FLXAnFRPRTC1 | 084C 0633 _H | $\langle \text{FLXAn_base} \rangle + 0090_{\text{H}}$ | 8, 16, 32 |
| FlexRay PRT Configuration Register 2 | FLXAnFRPRTC2 | 0F2D 0A0E _H | $\langle \text{FLXAn_base} \rangle + 0094_{\text{H}}$ | 8, 16, 32 |
| FlexRay MHD Configuration Register | FLXAnFRMHDC | 0000 0000 _H | $\langle \text{FLXAn_base} \rangle + 0098_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 1 | FLXAnFRGTUC1 | 0000 0280 _H | $\langle \text{FLXAn_base} \rangle + 00\text{A}0_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 2 | FLXAnFRGTUC2 | 0002 000A _H | $\langle \text{FLXAn_base} \rangle + 00\text{A}4_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 3 | FLXAnFRGTUC3 | 0202 0000 _H | $\langle \text{FLXAn_base} \rangle + 00\text{A}8_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 4 | FLXAnFRGTUC4 | 0008 0007 _H | $\langle \text{FLXAn_base} \rangle + 00\text{A}\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 5 | FLXAnFRGTUC5 | 0E00 0000 _H | $\langle \text{FLXAn_base} \rangle + 00\text{B}0_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 6 | FLXAnFRGTUC6 | 0002 0000 _H | $\langle \text{FLXAn_base} \rangle + 00\text{B}4_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 7 | FLXAnFRGTUC7 | 0002 0004 _H | $\langle \text{FLXAn_base} \rangle + 00\text{B}8_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 8 | FLXAnFRGTUC8 | 0000 0002 _H | $\langle \text{FLXAn_base} \rangle + 00\text{B}\text{C}_{\text{H}}$ | 8, 16, 32 |
| FlexRay GTU Configuration Register 9 | FLXAnFRGTUC9 | 0000 0101 _H | $\langle \text{FLXAn_base} \rangle + 00\text{C}0_{\text{H}}$ | 8, 16, 32 |

Table 18.8 FlexRay Register Map (2/3)

| Register Name | Symbol | Value after Reset | Address | Access Size |
|---|-------------------------------|------------------------|--|-------------|
| FlexRay GTU Configuration Register 10 | FLXAnFRGTUC10 | 0002 0005 _H | <FLXAn_base> + 00C4 _H | 8, 16, 32 |
| FlexRay GTU Configuration Register 11 | FLXAnFRGTUC11 | 0000 0000 _H | <FLXAn_base> + 00C8 _H | 8, 16, 32 |
| FlexRay CC Status Vector Register | FLXAnFRCCSV | 0010 4000 _H | <FLXAn_base> + 0100 _H | 8, 16, 32 |
| FlexRay CC Error Vector Register | FLXAnFRCCEV | 0000 0000 _H | <FLXAn_base> + 0104 _H | 8, 16, 32 |
| FlexRay Slot Counter Value Register | FLXAnFRSCV | 0000 0000 _H | <FLXAn_base> + 0110 _H | 8, 16, 32 |
| FlexRay Macrotick and Cycle Counter Value Register | FLXAnFRMTCCV | 0000 0000 _H | <FLXAn_base> + 0114 _H | 8, 16, 32 |
| FlexRay Rate Correction Value Register | FLXAnFRRCV | 0000 0000 _H | <FLXAn_base> + 0118 _H | 8, 16, 32 |
| FlexRay Offset Correction Value Register | FLXAnFROCV | 0000 0000 _H | <FLXAn_base> + 011C _H | 8, 16, 32 |
| FlexRay Sync Frame Status Register | FLXAnFRSFS | 0000 0000 _H | <FLXAn_base> + 0120 _H | 8, 16, 32 |
| FlexRay Symbol Window and NIT Status Register | FLXAnFRSWNIT | 0000 0000 _H | <FLXAn_base> + 0124 _H | 8, 16, 32 |
| FlexRay Aggregated Channel Status Register | FLXAnFRACS | 0000 0000 _H | <FLXAn_base> + 0128 _H | 8, 16, 32 |
| FlexRay Even Sync ID Register m (m = 1 to 15) | FLXAnFRESIDm (m = 1 to 15) | 0000 0000 _H | <FLXAn_base> + 0130 _H to <FLXAn_base> + 0168 _H (<FLXAn_base> + 0130 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay Odd Sync ID Register m (m = 1 to 15) | FLXAnFROSIDm (m = 1 to 15) | 0000 0000 _H | <FLXAn_base> + 0170 _H to <FLXAn_base> + 01A8 _H (<FLXAn_base> + 0170 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay Network Management Vector Register m (m = 1 to 3) | FLXAnFRNMVm (m = 1 to 3) | 0000 0000 _H | <FLXAn_base> + 01B0 _H to <FLXAn_base> + 01B8 _H (<FLXAn_base> + 01B0 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay Message RAM Configuration Register | FLXAnFRMRC | 0180 0000 _H | <FLXAn_base> + 0300 _H | 8, 16, 32 |
| FlexRay FIFO Rejection Filter Register | FLXAnFRFRF | 0180 0000 _H | <FLXAn_base> + 0304 _H | 8, 16, 32 |
| FlexRay FIFO Rejection Filter Mask Register | FLXAnFRFRFM | 0000 0000 _H | <FLXAn_base> + 0308 _H | 8, 16, 32 |
| FlexRay FIFO Critical Level Register | FLXAnFRFCL | 0000 0080 _H | <FLXAn_base> + 030C _H | 8, 16, 32 |
| FlexRay Message Handler Status Register | FLXAnFRMHDS | 0000 0080 _H | <FLXAn_base> + 0310 _H | 8, 16, 32 |
| FlexRay Last Dynamic Transmit Slot Register | FLXAnFRLDTS | 0000 0000 _H | <FLXAn_base> + 0314 _H | 8, 16, 32 |
| FlexRay FIFO Status Register | FLXAnFRFSR | 0000 0000 _H | <FLXAn_base> + 0318 _H | 8, 16, 32 |
| FlexRay Message Handler Constraints Flags Register | FLXAnFRMHDF | 0000 0000 _H | <FLXAn_base> + 031C _H | 8, 16, 32 |
| FlexRay Transmission Register m (m = 1 to 4) | FLXAnFRTXRQi (i = 1 to 4) | 0000 0000 _H | <FLXAn_base> + 0320 _H to <FLXAn_base> + 032C _H (<FLXAn_base> + 0320 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay New Data Register m (m = 1 to 4) | FLXAnFRNDATm (m = 1 to 4) | 0000 0000 _H | <FLXAn_base> + 0330 _H to <FLXAn_base> + 033C _H (<FLXAn_base> + 0330 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay Message Buffer Status Changed Register m (m = 1 to 4) | FLXAnFRMBSCm (m = 1 to 4) | 0000 0000 _H | <FLXAn_base> + 0340 _H to <FLXAn_base> + 034C _H (<FLXAn_base> + 0340 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay Write Data Section Register m (m = 1 to 64) | FLXAnFRWRDSm (m = 1 to 64) | 0000 0000 _H | <FLXAn_base> + 0400 _H to <FLXAn_base> + 04FC _H (<FLXAn_base> + 0400 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay Write Header Section Register 1 | FLXAnFRWRHS1 | 0000 0000 _H | <FLXAn_base> + 0500 _H | 8, 16, 32 |
| FlexRay Write Header Section Register 2 | FLXAnFRWRHS2 | 0000 0000 _H | <FLXAn_base> + 0504 _H | 8, 16, 32 |
| FlexRay Write Header Section Register 3 | FLXAnFRWRHS3 | 0000 0000 _H | <FLXAn_base> + 0508 _H | 8, 16, 32 |

Table 18.8 FlexRay Register Map (3/3)

| Register Name | Symbol | Value after Reset | Address | Access Size |
|---|-------------------------------|------------------------|--|-------------|
| FlexRay Input Buffer Command Mask Register | FLXAnFRIBCM | 0000 0000 _H | <FLXAn_base> + 0510 _H | 8, 16, 32 |
| FlexRay Input Buffer Command Request Register | FLXAnFRIBCR | 0000 0000 _H | <FLXAn_base> + 0514 _H | 8, 16, 32 |
| FlexRay Read Data Section Register m (m = 1 to 64) | FLXAnFRRDDSm (m = 1 to 64) | 0000 0000 _H | <FLXAn_base> + 0600 _H to <FLXAn_base> + 06FC _H (<FLXAn_base> + 0600 _H + (m-1) × 4) | 8, 16, 32 |
| FlexRay Read Header Section Register 1 | FLXAnFRRDHS1 | 0000 0000 _H | <FLXAn_base> + 0700 _H | 8, 16, 32 |
| FlexRay Read Header Section Register 2 | FLXAnFRRDHS2 | 0000 0000 _H | <FLXAn_base> + 0704 _H | 8, 16, 32 |
| FlexRay Read Header Section Register 3 | FLXAnFRRDHS3 | 0000 0000 _H | <FLXAn_base> + 0708 _H | 8, 16, 32 |
| FlexRay Message Buffer Status Register | FLXAnFRMBS | 0000 0000 _H | <FLXAn_base> + 070C _H | 8, 16, 32 |
| FlexRay Output Buffer Command Mask Register | FLXAnFROBCM | 0000 0000 _H | <FLXAn_base> + 0710 _H | 8, 16, 32 |
| FlexRay Output Buffer Command Request Register | FLXAnFROBCR | 0000 0000 _H | <FLXAn_base> + 0714 _H | 8, 16, 32 |
| FlexRay Input Transfer Configuration Register | FLXAnFRITC | 0000 0000 _H | <FLXAn_base> + 0800 _H | 8, 16, 32 |
| FlexRay Output Transfer Configuration Register | FLXAnFROTC | 0000 0000 _H | <FLXAn_base> + 0804 _H | 8, 16, 32 |
| FlexRay Input pointer table Base Address Register | FLXAnFRIBA | 0000 0000 _H | <FLXAn_base> + 0808 _H | 8, 16, 32 |
| FlexRay FIFO pointer table Base Address Register | FLXAnFRFBA | 0000 0000 _H | <FLXAn_base> + 080C _H | 8, 16, 32 |
| FlexRay Output pointer table Base Address Register | FLXAnFROBA | 0000 0000 _H | <FLXAn_base> + 0810 _H | 8, 16, 32 |
| FlexRay Input Queue Control Register | FLXAnFRIQC | 0000 0000 _H | <FLXAn_base> + 0814 _H | 8, 16, 32 |
| FlexRay User Input transfer Request Register | FLXAnFRUIR | 0000 0000 _H | <FLXAn_base> + 0818 _H | 8, 16, 32 |
| FlexRay User Output transfer Request Register | FLXAnFRUOR | 0000 0000 _H | <FLXAn_base> + 081C _H | 8, 16, 32 |
| FlexRay Input Transfer Status Register | FLXAnFRITS | 0000 0000 _H | <FLXAn_base> + 0820 _H | 8, 16, 32 |
| FlexRay Output Transfer Status Register | FLXAnFROTS | 0000 0000 _H | <FLXAn_base> + 0824 _H | 8, 16, 32 |
| FlexRay Access Error Status Register | FLXAnFRAES | 0000 0000 _H | <FLXAn_base> + 0828 _H | 8, 16, 32 |
| FlexRay Access Error Address Register | FLXAnFRAEA | 0000 0000 _H | <FLXAn_base> + 082C _H | 8, 16, 32 |
| FlexRay message Data Available Register m (m = 0 to 3) | FLXAnFRDAm (m = 0 to 3) | 0000 0000 _H | <FLXAn_base> + 0830 _H to <FLXAn_base> + 083C _H (<FLXAn_base> + 0830 _H + (m × 4)) | 8, 16, 32 |
| FlexRay H-Bus Configuration Register | FLXAnFRAHBC | 0000 0000 _H | <FLXAn_base> + 0840 _H | 8, 16, 32 |
| FlexRay Timer 2 Configuration Register | FLXAnFRT2C | 0000 0000 _H | <FLXAn_base> + 0844 _H | 8, 16, 32 |

18.2.2 FlexRay Operation register

18.2.2.1 FLXAnFROC — FlexRay Operation Control Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0004_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|-----|----|----|----|----|------|------|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | T2IE | T1IE | T0IE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | OEP | — | — | — | — | — | — | OE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R/W |

Table 18.9 FLXAnFROC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | T2IE | Timer 2 interrupt enable Bit 0: Disabled 1: Enabled |
| 17 | T1IE | Timer 1 interrupt enable Bit 0: Disabled 1: Enabled |
| 16 | T0IE | Timer 0 interrupt enable Bit 0: Disabled 1: Enabled |
| 15 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 | OEP | Operation Enable bit Protection Bit 0: OE is unprotected 1: OE is protected |
| 6 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | OE | Operation Enable Bit 0: Operation disabled, software reset of the FlexRay module 1: Operation enabled |

(1) FLXAnFROC.T2IE

Timer 2 interrupt enable bit

This bit controls the timer 2 interrupt.

0: Disabled

No interrupt will be requested and the timer 2 interrupt line will be released if pending.

1: Enabled

Timer 2 interrupt will be asserted when FLXAnFROTS.T2IS is 1.

(2) FLXAnFROC.T1IE

Timer 1 interrupt enable bit

The user should only set this bit to 1 when timer 1 interrupt is not enabled in the FlexRay Status interrupt enable register (bit FLXAnFRSIES.T1IE should be 0).

This bit controls the timer 1 interrupt.

0: Disabled

No interrupt will be requested and the timer 1 interrupt line will be released if pending.

1: Enabled

Timer 1 interrupt will be asserted when bit FLXAnFROTS.T1IS is 1.

(3) FLXAnFROC.T0IE

Timer 0 interrupt enable bit

The user should only set this bit to 1 when timer 0 interrupt is not enabled in the FlexRay Status interrupt enable register (bit FLXAnFRSIES.T0IE should be 0).

This bit controls the timer 0 interrupt.

0: Disabled

No interrupt will be requested and the timer 0 interrupt line will be released if pending.

1: Enabled

Timer 0 interrupt will be asserted when bit FLXAnFROTS.T0IS is 1.

(4) FLXAnFROC.OEP

Operation enable bit protection bit

This bit protects against unintended write access to the OE bit.

0: OE bit is unprotected

Write access to the OE bit is enabled

1: OE bit is protected

Write access to the OE bit is disabled

(5) FLXAnFROC.OE

Operation enable bit

The user can only write to this bit when bit FLXAnFROC.OEP is 0.

The user should only write this bit with 0 when bit FLXAnFROS.OS is 1.

The user should only write this bit with 1 when bit FLXAnFROS.OS is 0 and the FlexRay sample clock is enabled.

This bit controls the operation state and serves the software reset of the FlexRay module. The operation status bit (FLXAnFROS.OS) indicates whether the FlexRay module is in reset state or not.

0: Operation disabled, software reset of the FlexRay module

Forcibly moves the FlexRay module to its reset state, whatever the state of the FlexRay module is.

1: Operation enabled

Reset state of the FlexRay module is released.

18.2.2.2 FLXAnFROS — FlexRay Operation Status Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 000C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | T2IS | T1IS | T0IS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.10 FLXAnFROS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | T2IS | Timer 2 Interrupt Status Bit 0: Timer 2 has not matched the conditions configured in the FLXAnFRT2C register 1: Timer 2 matched the conditions configured in the FLXAnFRT2C register |
| 17 | T1IS | Timer 1 Interrupt Status Bit 0: Timer 1 has not matched the conditions configured in the FLXAnFRT1C register 1: Timer 1 matched the conditions configured in the FLXAnFRT1C register |
| 16 | T0IS | Timer 0 Interrupt Status Bit 0: Timer 0 has not matched the conditions configured in the FLXAnFRT0C register 1: Timer 0 matched the conditions configured in the FLXAnFRT0C register |
| 15 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | OS | Operation Status Bit 0: Operation disabled, reset state 1: Operation enabled |

(1) FLXAnFROS.T2IS

Timer 2 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the timer 2 has matched the state configured in the FLXAnFRT2C register.

When the FLXAnFROC.T2IE bit is enabled and the FLXAnFROS.T2IS bit is set to 1, the timer 2 interrupt is generated.

[Clearing condition]

- This bit is cleared by writing 1 to the FLXAnFROS.T2IS bit.
- This bit is cleared when the FLXAnFROS.OS bit changes from 1 to 0.

[Setting condition]

- This bit is set when the state matches the state configured in the FLXAnFRT2C register.

(2) FLXAnFROS.T1IS

Timer 1 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the timer 1 has matched the state configured in the FLXAnFRT1C register.

When the FLXAnFROC.T1IE bit is enabled and the FLXAnFROS.T1IS bit is set to 1, the timer 1 interrupt is generated.

[Clearing condition]

- This bit is cleared by writing 1 to the FLXAnFROS.T1IS bit.
This bit is cleared when the FLXAnFROS.OS bit changes from 1 to 0.

[Setting condition]

- This bit is set when the state becomes the state configured in the FLXAnFRT1C register.

(3) FLXAnFROS.T0IS

Timer 0 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the timer 0 has matched the state configured in the FLXAnFRT0C register.

When the FLXAnFROC.T0IE bit is enabled and the FLXAnFROS.T0IS bit is set to 1, the timer 0 interrupt is generated.

[Clearing condition]

- This bit is cleared by writing 1 to the FLXAnFROS.T0IS bit.
- This bit is cleared when the FLXAnFROS.OS bit changes from 1 to 0.

[Setting condition]

- This bit is set when the state becomes the state configured in the FLXAnFRT0C register.

(4) FLXAnFROS.OS

Operation Status Bit

This bit represents if the FlexRay module is in the reset or the operation state.

When bit FLXAnFROS.OS is 0 the FlexRay module gets initialized and registers mapped to the address area $\langle \text{FLXAn_base} \rangle + 0010_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 0\text{FFF}_{\text{H}}$ cannot be accessed; read access from these registers will return undefined data.

When bit FLXAnFROS.OS is 1 it is possible to access to the address area $\langle \text{FLXAn_base} \rangle + 0010_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 0\text{FFF}_{\text{H}}$ and to perform FlexRay communication.

When bit FLXAnFROS.OS changes from 0 to 1 all registers in the address range $\langle \text{FLXAn_base} \rangle + 0010_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 0\text{FFF}_{\text{H}}$ are set to the “Values after reset”.

[Clearing condition]

- When bit FLXAnFROC.OE is set to 0. It takes up to two peripheral bus clock cycles until bit FLXAnFROS.OS is set to 0.

[Setting condition]

- When bit FLXAnFROC.OE is set to 1 it takes up to four peripheral clock cycles of the clock with the lower frequency out of the FlexRay sample clock and peripheral bus clock until bit FLXAnFROS.OS is set to 1.

18.2.3 Special Registers

18.2.3.1 FLXAnFRLCK — FlexRay Lock Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 001C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | | | | | | | | CLK[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.11 FLXAnFRLCK Register Contents

| Bit | Symbol | Function |
|---------|----------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 to 0 | CLK[7:0] | Configuration Lock Key Bit |

(1) FLXAnFRLCK.CLK

Configuration Lock Key Bit

The Lock Register is write-only. Reading the register will return 0000 0000_H.

To leave CONFIG state by writing bits FLXAnFRSUCC1.CMD[3:0] (command READY), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the FLXAnFRSUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated.

First write: Bits FLXAnFRLCK.CLK[7:0] = “1100 1110_B” (CE_H)

Second write: Bits FLXAnFRLCK.CLK[7:0] = “0011 0001_B” (31_H)

Third write: Bits FLXAnFRSUCC1.CMD[3:0]

CAUTION

In case that the Host uses 8/16-bit accesses to write the listed bit fields, the user has to ensure that no “dummy accesses” e.g. to the remaining register bytes / words are inserted by the compiler.

18.2.4 Interrupt Registers

18.2.4.1 FLXAnFREIR — FlexRay Error Interrupt Register

The flags are set when the CC detects one of the listed error conditions. The flags remain set until the Host clears them.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0020_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|------|------|-----|----|----|----|----|----|------|------|-----|
| | — | — | — | — | — | TABB | LTVB | EDB | — | — | — | — | — | TABA | LTVA | EDA |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|-----|------|------|-----|-----|------|-----|-----|-----|------|-----|------|
| | — | — | — | — | MHF | IOBA | IIBA | EFA | RFO | AERR | CCL | CCF | SFO | SFBM | CNA | PEMC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.12 FLXAnFREIR Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 | TABB | Transmission Across Boundary Channel B Flag 0: No transmission across slot boundary detected on channel B 1: Transmission across slot boundary detected on channel B |
| 25 | LTVB | Latest Transmit Violation Channel B Flag 0: No latest transmit violation detected on channel B 1: Latest transmit violation detected on channel B |
| 24 | EDB | Error Detected on Channel B Flag 0: No error detected on channel B 1: Error detected on channel B |
| 23 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | TABA | Transmission Across Boundary Channel A Flag 0: No transmission across slot boundary detected on channel A 1: Transmission across slot boundary detected on channel A |
| 17 | LTVA | Latest Transmit Violation Channel A Flag 0: No latest transmit violation detected on channel A 1: Latest transmit violation detected on channel A |
| 16 | EDA | Error Detected on Channel A Flag 0: No error detected on channel A 1: Error detected on channel A |
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 | MHF | Message Handler Constraints Flag 0: No Message Handler failure detected 1: Message Handler failure detected |
| 10 | IOBA | Illegal Output buffer Access Flag 0: No illegal Host access to Output Buffer occurred 1: Illegal Host access to Output Buffer occurred |

Table 18.12 FLXAnFREIR Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 9 | IIBA | Illegal Input Buffer Access Flag 0: No illegal Host access to Input Buffer occurred 1: Illegal Host access to Input Buffer occurred |
| 8 | EFA | Empty FIFO Access Flag 0: No Host access to empty FIFO occurred 1: Host access to empty FIFO occurred |
| 7 | RFO | Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has occurred |
| 6 | AERR | Access error flag Flag 0: Access error is not detected. 1: Access error is detected. |
| 5 | CCL | CHI Command Locked Flag 0: CHI command accepted 1: CHI command not accepted |
| 4 | CCF | Clock Correction Failure Flag 0: No clock correction error 1: Clock correction failed |
| 3 | SFO | Sync Frame Overflow Flag 0: Number of received sync frames \leq the FLXAnFRGTUC2.SNM bit 1: More sync frames received than configured by the FLXAnFRGTUC2.SNM bit |
| 2 | SFBM | Sync Frames Below Minimum Flag 0: Sync node: 1 or more sync frames received Non-sync node: 2 or more sync frames received 1: Less than the required minimum of sync frames received |
| 1 | CNA | Command Not Accepted Flag 0: CHI command accepted 1: CHI command not accepted |
| 0 | PEMC | POC Error Mode Changed Flag 0: Error mode has not changed 1: Error mode has changed |

(1) FLXAnFREIR.TABB

Transmission Across Boundary Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when 1 is written.

The flag signals to the Host that a transmission across a slot boundary occurred for channel B.

(2) FLXAnFREIR.LTVB

Latest Transmit Violation Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals a latest transmit violation on channel B to the Host.

(3) FLXAnFREIR.EDB

Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This bit is set whenever one of bits FLXAnFRACS.SEDB, FLXAnFRACS.CEDB, FLXAnFRACS.CIB, and FLXAnFRACS.SBVB changes from 0 to 1.

(4) FLXAnFREIR.TABA

Transmission Across Boundary Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals to the Host that a transmission across a slot boundary occurred for channel A.

(5) FLXAnFREIR.LTVA

Latest Transmit Violation Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals a latest transmit violation on channel A to the Host.

(6) FLXAnFREIR.EDA

Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This bit is set whenever one of bits FLXAnFRACS.SEDA, FLXAnFRACS.CEDA, FLXAnFRACS.CIA, and FLXAnFRACS.SBVA changes from 0 to 1.

(7) FLXAnFREIR.MHF

Message Handler Constraints Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals a Message Handler constraints violation condition. It is set whenever one of the bits FLXAnFRMHDF.SNUA, FLXAnFRMHDF.SNUB, FLXAnFRMHDF.FNFA, FLXAnFRMHDF.FNFB, FLXAnFRMHDF.TBFA, FLXAnFRMHDF.TBFB, and FLXAnFRMHDF.WAHP changes from 0 to 1.

(8) FLXAnFREIR.IOBA

Illegal Output buffer Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while bit FLXAnFROBCR.OBSYS is set to 1.

(9) FLXAnFREIR.IIBA

Illegal Input Buffer Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Host wants to modify a message buffer via Input Buffer and one of the following conditions applies:

1. The CC is not in CONFIG or DEFAULT_CONFIG state and the Host writes to the Input Buffer Command Request register to modify the
 - Header section of message buffer 0, 1 if configured for transmission in key slot
 - Header section of static message buffers with buffer number < FLXAnFRMRC.FDB[7:0] while bits FLXAnFRMRC.SEC[1:0] = “01”
 - Header section of any static or dynamic message buffer while bits FLXAnFRMRC.SEC[1:0] = “1x”
 - Header and / or data section of any message buffer belonging to the receive FIFO
2. The Host writes to any register of the Input Buffer while bit FLXAnFRIBCR.IBSYH is set to 1.

(10) FLXAnFREIR.EFA

Empty FIFO Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty.

(11) FLXAnFREIR.RFO

Receive FIFO Overrun Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register FLXAnFRFSR.

(12) FLXAnFREIR.AERR

Access error flag Flag

Writing 0 in this bit has no effect.

This bit is cleared when writing 1 to it.

Notifies of an access error.

When bit FLXAnFRMHDS.AMR, FLXAnFRMHDS.ATBF1, or FLXAnFRMHDS.ATBF2 changes from 0 to 1, this bit is set to 1.

(13) FLXAnFREIR.CCL

CHI Command Locked Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals that the write access to the CHI command vector FLXAnFRSUCC1.CMD[3:0] bits was not successful because the execution of the previous CHI command has not yet completed. In this case bit FLXAnFREIR.CNA is also set to 1.

(14) FLXAnFREIR.CCF

Clock Correction Failure Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set at the end of the communication cycle whenever one of the following errors occurred:

- Missing offset and / or rate correction
- Clock correction limit reached

The clock correction status is monitored in registers FLXAnFRCCEV and FLXAnFRSFS. A failure may occur during startup, therefore bit FLXAnFREIR.CCF should be set to 0 after the CC entered NORMAL_ACTIVE state.

(15) FLXAnFREIR.SFO

Sync Frame Overflow Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

It is set to 1 when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last double cycle exceeds the maximum number of sync frames as defined by the FLXAnFRGTUC2.SNM[3:0] bits.

(16) FLXAnFREIR.SFBM

Sync Frames Below Minimum Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 at the end of a cycle if the number of sync frames received during the last communication cycle was below the limit required for rate or offset correction term calculation (i.e. missing offset and / or missing rate correction). The clock correction status is monitored in FLXAnFRCCEV and FLXAnFRSFS.

This flag may be set to 1 during startup. Therefore this flag should be set to 0 by the Host after the CC entered NORMAL_ACTIVE state.

(17) FLXAnFREIR.CNA

Command Not Accepted Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals that the write access to the CHI command vector FLXAnFRSUCC1.CMD[3:0] bits was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (FLXAnFREIR.CCL = 1).

(18) FLXAnFREIR.PEMC

POC Error Mode Changed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 whenever the error mode signaled by the FLXAnFRCCEV.ERRM[1:0] bits has changed.

18.2.4.2 FLXAnFRSIR — FlexRay Status Interrupt Register

The flags are set when the CC detects one of the listed events. The flags remain set until the Host clears them.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0024_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----|------|------|-----|------|------|------|------|------|------|------|-----|-----|------|------|------|
| | — | — | — | — | — | — | MTSB | WUPB | — | — | — | — | — | — | MTSA | WUPA |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SDS | MBSI | SUCS | SWE | TOBC | TIBC | TI1 | TI0 | NMVC | RFCL | RFNE | RXI | TXI | CYCS | CAS | WST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.13 FLXAnFRSIR Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 26 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 25 | MTSB | MTS Received on Channel B Flag (vSS!ValidMTSB) 0: No MTS symbol received on channel B 1: MTS symbol received on channel B |
| 24 | WUPB | Wakeup Pattern Channel B Flag 0: No wakeup pattern received on channel B 1: Wakeup pattern received on channel B |
| 23 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17 | MTSA | MTS Received on Channel A Flag (vSS!ValidMTSA) 0: No MTS symbol received on channel A 1: MTS symbol received on channel A |
| 16 | WUPA | Wakeup Pattern Channel A Flag 0: No wakeup pattern received on channel A 1: Wakeup pattern received on channel A |
| 15 | SDS | Start of Dynamic Segment Flag 0: Dynamic segment not yet started 1: Dynamic segment started |
| 14 | MBSI | Message Buffer Status Interrupt Flag 0: No message buffer status change of message buffer with MBI = 1 1: Message buffer status of at least one message buffer with MBI = 1 has changed |
| 13 | SUCS | Startup Completed Successfully Flag 0: No startup completed successfully 1: Startup completed successfully |
| 12 | SWE | Stop Watch Event Flag 0: No Stop Watch Event 1: Stop Watch Event occurred |
| 11 | TOBC | Transfer Output Buffer Completed Flag 0: No transfer completed 1: Transfer between Message RAM and Output Buffer completed |

Table 18.13 FLXAnFRSIR Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 10 | TIBC | Transfer Input Buffer Completed Flag 0: No transfer completed 1: Transfer between Input Buffer and Message RAM completed |
| 9 | TI1 | Timer Interrupt 1 Flag 0: No timer interrupt 1 1: Timer interrupt 1 occurred |
| 8 | TI0 | Timer Interrupt 0 Flag 0: No timer interrupt 0 1: Timer interrupt 0 occurred |
| 7 | NMVC | Network Management Vector Changed Flag 0: No change in the network management vector 1: Network management vector changed |
| 6 | RFCL | Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached |
| 5 | RFNE | Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty |
| 4 | RXI | Receive Interrupt Flag 0: No ND flag of a receive buffer with MBI = 1 has been set to 1 1: At least one ND flag of a receive buffer with MBI = 1 has been set to 1 |
| 3 | TXI | Transmit Interrupt Flag 0: No frame transmitted from a transmit buffer with MBI = 1 1: At least one frame was transmitted from a transmit buffer with MBI = 1 |
| 2 | CYCS | Cycle Start Interrupt Flag 0: No communication cycle started 1: Communication cycle started |
| 1 | CAS | Collision Avoidance Symbol Flag 0: No bit pattern matching the CAS symbol received 1: Bit pattern matching the CAS symbol received |
| 0 | WST | Wakeup Status Flag 0: Wakeup status unchanged 1: Wakeup status changed |

(1) FLXAnFRSIR.MTSB

MTS Received on Channel B Flag (vSS!ValidMTSB)

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Media Access Test symbol received on channel B during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(2) FLXAnFRSIR.WUPB

Wakeup Pattern Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 when a wakeup pattern was received on channel B in either of the following states:

- WAKEUP
- READY
- STARTUP

(3) FLXAnFRSIR.MTSA

MTS Received on Channel A Flag (vSS!ValidMTSA)

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Media Access Test symbol received on channel A during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(4) FLXAnFRSIR.WUPA

Wakeup Pattern Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 when a wakeup pattern was received on channel A in either of the following states:

- WAKEUP
- READY
- STARTUP

(5) FLXAnFRSIR.SDS

Start of Dynamic Segment Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the dynamic segment starts.

(6) FLXAnFRSIR.MBSI

Message Buffer Status Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the message buffer status FLXAnFRMBS register has changed and if bit MBI of that message buffer is 1 (see **Table 18.106**).

(7) FLXAnFRSIR.SUCS

Startup Completed Successfully Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state.

(8) FLXAnFRSIR.SWE

Stop Watch Event Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the Stop Watch register (see **Section 18.2.5.4, FLXAnFRSTPW1 — FlexRay Stop Watch Register 1**).

(9) FLXAnFRSIR.TOBC

Transfer Output Buffer Completed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever a transfer from the Message RAM to the Output Buffer has completed and bit FLXAnFROBCR.OBSYS has been reset by the Message Handler.

(10) FLXAnFRSIR.TIBC

Transfer Input Buffer Completed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever a transfer from Input Buffer to the Message RAM has completed and bit FLXAnFRIBCR.IBSYS has been reset by the Message Handler.

(11) FLXAnFRSIR.TI1

Timer 1 Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever timer 1 matches the conditions configured in register FLXAnFRT1C.

FlexRay timer 1 interrupt is generated when bit FLXAnFROC.T1IE is effective.

(12) FLXAnFRSIR.TI0

Timer 0 Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever timer 0 matches the conditions configured in register FLXAnFRT0C.

FlexRay timer 0 interrupt is generated when bit FLXAnFROC.T0IE is effective.

(13) FLXAnFRSIR.NMVC

Network Management Vector Changed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This is set when a change in the Network Management Vector occurs.

(14) FLXAnFRSIR.RFCL

Receive FIFO Critical Level Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set when the receive FIFO fill level indicated by bits FLXAnFRFSR.RFLL[7:0] is equal or greater than the critical level as configured by bit FLXAnFRFCL.CL.

(15) FLXAnFRSIR.RFNE

Receive FIFO Not Empty Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when a received valid frame was stored into the empty receive FIFO. The actual state of the receive FIFO is monitored in register FLXAnFRFSR.

(16) FLXAnFRSIR.RXI

Receive Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC whenever the set condition of a message buffers ND flag is fulfilled (see **Section 18.2.9.6, FLXAnFRNDATm — FlexRay New Data Register m (m = 1 to 4)**, and if bit MBI of that message buffer is set to 1 (see **Table 18.106**))

(17) FLXAnFRSIR.TXI

Transmit Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC at the end of frame transmission if bit MBI in the respective message buffer is set to 1 (see **Table 18.106**).

(18) FLXAnFRSIR.CYCS

Cycle Start Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when a communication cycle starts.

(19) FLXAnFRSIR.CAS

Collision Avoidance Symbol Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC during STARTUP state when a CAS or a potential CAS was received.

(20) FLXAnFRSIR.WST

Wakeup Status Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set when flags FLXAnFRCCSV.WSV[2:0] change to a value other than UNDEFINED.

18.2.4.3 FLXAnFREILS — FlexRay Error Interrupt Line Select Register

This register assigns an interrupt generated by a specific error interrupt flag from register FLXAnFREIR to one of the two modules interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0028_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|------|-------|-------|------|------|-------|------|------|------|-------|-------|-------|
| | — | — | — | — | — | TABBL | LTVBL | EDBL | — | — | — | — | — | TABAL | LTVAL | EDAL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | MHFL | IOBAL | IIBAL | EFAL | RFOL | AERRL | CCLL | CCFL | SFOL | SFBML | CNAL | PEMCL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.14 FLXAnFREILS Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 | TABBL | Transmission Across Boundary Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 25 | LTVBL | Latest Transmit Violation Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 24 | EDBL | Error Detected on Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 23 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | TABAL | Transmission Across Boundary Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 17 | LTVAL | Latest Transmit Violation Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 16 | EDAL | Error Detected on Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 | MHFL | Message Handler Constraints Flag Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 10 | IOBAL | Illegal Output Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 9 | IIBAL | Illegal Input Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |

Table 18.14 FLXAnFREILS Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 8 | EFAL | Empty FIFO Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 7 | RFOL | Receive FIFO Overrun Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 6 | AERRL | Access Error Interrupt Output Select Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 5 | CCLL | CHI Command Locked Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 4 | CCFL | Clock Correction Failure Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 3 | SFOL | Sync Frame Overflow Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 2 | SFBML | Sync Frames Below Minimum Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 1 | CNAL | Command Not Accepted Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 0 | PEMCL | POC Error Mode Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |

18.2.4.4 FLXAnFRSILS — FlexRay Status Interrupt Line Select Register

This register assigns an interrupt generated by a specific status interrupt flag from register FLXAnFRSIR to one of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 002C_H

Value after reset: 0303 FFFF_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|-------|-------|------|-------|-------|-------|-------|-------|-------|-------|------|------|-------|-------|-------|
| | — | — | — | — | — | — | MTSBL | WUPBL | — | — | — | — | — | — | MTSAL | WUPAL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SDSL | MBSIL | SUCSL | SWEL | TOBCL | TIBCL | TI1L | TI0L | NMVCL | RFCLL | RFNEL | RXIL | TXIL | CYCSL | CASL | WSTL |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.15 FLXAnFRSILS Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 26 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 25 | MTSBL | Media Access Test Symbol Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 24 | WUPBL | Wakeup Pattern Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 23 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17 | MTSAL | Media Access Test Symbol Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 16 | WUPAL | Wakeup Pattern Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 15 | SDSL | Start of Dynamic Segment Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 14 | MBSIL | Message Buffer Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 13 | SUCSL | Startup Completed Successfully Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 12 | SWEL | Stop Watch Event Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 11 | TOBCL | Transfer Output Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |

Table 18.15 FLXAnFRSILS Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 10 | TIBCL | Transfer Input Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 9 | TI1L | Timer 1 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 8 | TI0L | Timer 0 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 7 | NMVCL | Network Management Vector Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 6 | RFCLL | Receive FIFO Critical Level Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 5 | RFNEL | Receive FIFO Not Empty Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 4 | RXIL | Receive Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 3 | TXIL | Transmit Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 2 | CYCSL | Cycle Start Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 1 | CASL | Collision Avoidance Symbol Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |
| 0 | WSTL | Wakeup Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt |

18.2.4.5 FLXAnFREIES — FlexRay Error Interrupt Enable Set Register

The settings in the FlexRay Error Interrupt Enable Set (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 sets the interrupt enable bit.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0030_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|------|-------|-------|------|------|-------|------|------|------|-------|-------|-------|
| | — | — | — | — | — | TABBE | LTVBE | EDBE | — | — | — | — | — | TABAE | LTVAE | EDAE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | MHFE | IOBAE | IIBAE | EFAE | RFOE | AERRE | CCLC | CCFE | SFOE | SFBME | CNAE | PEMCE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.16 FLXAnFREIES Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 | TABBE | Transmission Across Boundary Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 25 | LTVBE | Latest Transmit Violation Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 24 | EDBE | Error Detected on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 23 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | TABAE | Transmission Across Boundary Violation Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 17 | LTVAE | Latest Transmit Violation Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 16 | EDAE | Error Detected on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 18.16 FLXAnFREIES Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 11 | MHFE | Message Handler Constraints Flag Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 10 | IOBAE | Illegal Output Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 9 | IIBAE | Illegal Input Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 8 | EFAE | Empty FIFO Access Interrupt Enable3 Bit 0: Interrupt disabled 1: Interrupt enabled |
| 7 | RFOE | Receive FIFO Overrun Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 6 | AERRE | Access Error Interrupt Enable Bit 0: Interrupt is disabled. 1: Interrupt is enabled. |
| 5 | CCLE | CHI Command Locked Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 4 | CCFE | Clock Correction Failure Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 3 | SFOE | Sync Frame Overflow Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 2 | SFBME | Sync Frames Below Minimum Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 1 | CNAE | Command Not Accepted Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 0 | PEMCE | POC Error Mode Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |

18.2.4.6 FLXAnFREIER — FlexRay Error Interrupt Enable Reset Register

The settings in the FlexRay Error Interrupt Enable Set (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 clears the interrupt enable bit.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0034_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|------|-------|-------|------|------|-------|------|------|------|-------|-------|-------|
| | — | — | — | — | — | TABBD | LTVBD | EDBD | — | — | — | — | — | TABAD | LTVAD | EDAD |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | MHFD | IOBAD | IIBAD | EFAD | RFOD | AERRD | CCLD | CCFD | SFOD | SFBMD | CNAD | PEMCD |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.17 FLXAnFREIER Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 | TABBD | Transmission Across Boundary Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 25 | LTVBD | Latest Transmit Violation Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 24 | EDBD | Error Detected on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 23 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | TABAD | Transmission Across Boundary Violation Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 17 | LTVAD | Latest Transmit Violation Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 16 | EDAD | Error Detected on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 18.17 FLXAnFREIER Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 11 | MHFD | Message Handler Constraints Flag Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 10 | IOBAD | Illegal Output Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 9 | IIBAD | Illegal Input Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 8 | EFAD | Empty FIFO Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 7 | RFOD | Receive FIFO Overrun Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 6 | AERRD | Access Error Interrupt Disable Bit 0: Interrupt is disabled. 1: Interrupt is enabled. |
| 5 | CCLD | CHI Command Locked Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 4 | CCFD | Clock Correction Failure Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 3 | SFOD | Sync Frame Overflow Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 2 | SFBMD | Sync Frames Below Minimum Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 1 | CNAD | Command Not Accepted Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 0 | PEMCD | POC Error Mode Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |

18.2.4.7 FLXAnFRSIES — FlexRay Status Interrupt Enable Set Register

The settings in the FlexRay Status Interrupt Enable Set (FLXAnFRSIES) and FlexRay Status Interrupt Enable Reset (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 sets the interrupt enable bit.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0038_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|-------|-------|------|-------|-------|-------|-------|-------|-------|-------|------|------|-------|-------|-------|
| | — | — | — | — | — | — | MTSBE | WUPBE | — | — | — | — | — | — | MTSAE | WUPAE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SDSE | MBSIE | SUCSE | SWEE | TOBCE | TIBCE | T1IE | TIOE | NMVCE | RFCLC | RFNEE | RXIE | TXIE | CYCSE | CASE | WSTE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.18 FLXAnFRSIES Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 26 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 25 | MTSBE | MTS Received on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 24 | WUPBE | Wakeup Pattern Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 23 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17 | MTSAE | MTS Received on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 16 | WUPAE | Wakeup Pattern Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 15 | SDSE | Start of Dynamic Segment Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 14 | MBSIE | Message Buffer Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 13 | SUCSE | Startup Completed Successfully Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |

Table 18.18 FLXAnFRSIES Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 12 | SWEE | Stop Watch Event Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 11 | TOBCE | Transfer Output Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 10 | TIBCE | Transfer Input Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 9 | TI1E | Timer 1 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 8 | TI0E | Timer 0 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 7 | NMVCE | Network Management Vector Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 6 | RFCLE | Receive FIFO Critical Level Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 5 | RFNEE | Receive FIFO Not Empty Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 4 | RXIE | Receive Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 3 | TXIE | Transmit Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 2 | CYCSE | Cycle Start Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 1 | CASE | Collision Avoidance Symbol Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 0 | WSTE | Wakeup Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled |

18.2.4.8 FLXAnFRSIER — FlexRay Status Interrupt Enable Reset Register

The settings in the FlexRay Status Interrupt Enable Set (FLXAnFRSIES) and FlexRay Status Interrupt Enable Reset (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 clears the interrupt enable bit.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 003C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|-------|-------|------|-------|-------|-------|-------|-------|-------|-------|------|------|-------|-------|-------|
| | — | — | — | — | — | — | MTSBD | WUPBD | — | — | — | — | — | — | MTSAD | WUPAD |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SDSD | MBSID | SUCSD | SWED | TOBCD | TIBCD | TI1D | TI0D | NMVCD | RFCLD | RFNED | RXID | TXID | CYCSD | CASD | WSTD |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.19 FLXAnFRSIER Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 26 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 25 | MTSBD | MTS Received on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 24 | WUPBD | Wakeup Pattern Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 23 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17 | MTSAD | MTS Received on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 16 | WUPAD | Wakeup Pattern Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 15 | SDSD | Start of Dynamic Segment Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 14 | MBSID | Message Buffer Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 13 | SUCSD | Startup Completed Successfully Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |

Table 18.19 FLXAnFRSIER Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 12 | SWED | Stop Watch Event Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 11 | TOBCD | Transfer Output Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 10 | TIBCD | Transfer Input Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 9 | TI1D | Timer Interrupt 1 Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 8 | TI0D | Timer Interrupt 0 Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 7 | NMVCD | Network Management Vector Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 6 | RFCLD | Receive FIFO Critical Level Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 5 | RFNED | Receive FIFO Not Empty Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 4 | RXID | Receive Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 3 | TXID | Transmit Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 2 | CYCSD | Cycle Start Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 1 | CASD | Collision Avoidance Symbol Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |
| 0 | WSTD | Wakeup Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled |

18.2.4.9 FLXAnFRILE — FlexRay Interrupt Line Enable Register

Each of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt) can be enabled / disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0040_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | EINT1 | EINT0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 18.20 FLXAnFRILE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | EINT1 | Enable FlexRay 1 Interrupt Line Bit 0: FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt enabled |
| 0 | EINT0 | Enable FlexRay 0 Interrupt Line Bit 0: FlexRay 0 interrupt disabled 1: FlexRay 0 interrupt enabled |

18.2.5 FlexRay Timer Registers

18.2.5.1 FLXAnFRT0C — FlexRay Timer 0 Configuration Register

This register is an absolute timer. It specifies the point in time when a FlexRay timer 0 interrupt occurs as the values of cycle count and macrotick (MT). When the FlexRay timer 0 passes, bits FLXAnFRSIR.TI0 and FLXAnFROS.TOIS are set to 1. A timer 0 interrupt then occurs while bit FLXAnFROC.TOIE is effective.

CAUTION

The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0044_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|-----------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| | — | — | T0MO[13:0] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | T0CC[6:0] | | | | | | — | — | — | — | — | — | — | T0MS | T0RC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R/W | R/W |

Table 18.21 FLXAnFRT0C Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 to 16 | T0MO[13:0] | Timer 0 Macrotick Offset Bit Timer 0 Macrotick Offset |
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 14 to 8 | T0CC[6:0] | Timer 0 Cycle Code Bit Timer 0 Cycle Code |
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | T0MS | Timer 0 Mode Select Bit 0: Single-shot mode 1: Continuous mode |
| 0 | T0RC | Timer 0 Run Control Bit 0: Timer 0 halted 1: Timer 0 running |

(1) FLXAnFRT0C.TOMO

Timer 0 Macrotick Offset Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT0C.TORC to 0.

Configures the macrotick offset from the beginning of the communication cycle where the interrupt is to occur. The FlexRay timer 0 interrupt occurs at this offset for each cycle of the cycle set.

(2) FLXAnFRT0C.T0CC

Timer 0 Cycle Code Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT0C.TORC to 0.

The 7-bit timer 0 cycle code determines the cycle set used for generation of the FlexRay timer 0 interrupt. For details about the configuration of the cycle code see **Section 18.3.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT0C.TOMS

Timer 0 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT0C.TORC to 0.

Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(4) FLXAnFRT0C.TORC

Timer 0 Run Control Bit

Timer 0 can be activated (set bit FLXAnFRT0C.TORC to 1) when the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

18.2.5.2 FLXAnFRT1C — FlexRay Timer 1 Configuration Register

This register is a relative timer. After the specified number of macroticks (MT) has expired, a FlexRay timer 1 interrupt is asserted. When the FlexRay timer 1 passes, bits FLXAnFRSIR.TI1 and FLXAnFROS.TIIS are set to 1. A timer 1 interrupt then occurs while bit FLXAnFROC.TIIE is effective.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0048_H

Value after reset: 0002 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | T1MC[13:0] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | | | | | | | | | | | | | | T1MS | T1RC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 18.22 FLXAnFRT1C Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 to 16 | T1MC[13:0] | Specify timer 1 macrotick count value. |
| 15 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | T1MS | Timer 1 Mode Select Bit 0: Single-shot mode 1: Continuous mode |
| 0 | T1RC | Timer 1 Run Control Bit 0: Timer 1 halted 1: Timer 1 running |

(1) FLXAnFRT1C.T1MC

Timer 1 Macrotick Count Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT1C.T1RC to 0.

Valid values are 2 to 16383 MT in continuous mode

Valid values are 1 to 16383 MT in single-shot mode

When the configured macrotick count is reached the FlexRay timer 1 interrupt is generated.

(2) FLXAnFRT1C.T1MS

Timer 1 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT1C.T1RC to 0.

Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(3) FLXAnFRT1C.T1RC

Timer 1 Run Control Bit

Timer 1 can be activated (set bit FLXAnFRT1C.T1RC to 1) as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

18.2.5.3 FLXAnFRT2C — FlexRay Timer 2 Configuration Register

This register is an absolute timer. Timer 2 has the same absolute timer features as timer 0.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0844_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|-----------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | T2MO[13:0] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | T2CC[6:0] | | | | | | — | — | — | — | — | — | — | T2MS | T2RC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R/W | R/W |

Table 18.23 FLXAnFRT2C Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 to 16 | T2MO[13:0] | Timer 2 Macrotick Offset Bit Timer 2 Macrotick Offset |
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 14 to 8 | T2CC[6:0] | Timer 2 Cycle Code Bit Timer 2 Cycle Code |
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | T2MS | Timer 2 Mode Select Bit 0: Single-shot mode 1: Continuous mode |
| 0 | T2RC | Timer 2 Run Control Bit 0: Timer halted 1: Timer running |

(1) FLXAnFRT2C.T2MO

Timer 2 Macrotick Offset Bit

Stop the timer by writing 0 to the FLXAnFRT2C.T2RC bit before changing the setting of the timer.

The T2MO bits are used to set the timing for generation of the timer 2 interrupt as an offset value in MT units from the position where the transfer cycle starts. That is, the timer 2 interrupt is generated at the offset position specified by the setting for the number of MT cycles.

(2) FLXAnFRT2C.T2CC

Timer 2 Cycle Code Bit

Stop the timer by writing 0 to bit FLXAnFRT2C.T2RC before changing the setting of the timer.

The T2CC bits are used to make the cycle setting for generation of the timer 2 interrupt as a 7-bit timer 2 cycle code. For details, see **Section 18.3.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT2C.T2MS

Timer 2 Mode Select Bit

Stop the timer by writing 0 to bit FLXAnFRT2C.T2RC before changing the setting of the timer.

Set the execution mode of the timer. In single shot mode, the timer stops when the timer setting matched with the MT value of the cycle counter.

(4) FLXAnFRT2C.T2RC

Timer 2 Run Control Bit

The timer 2 can operate only when POC is in NORMAL_ACTIVE state or NORMAL_PASSIVE state (bit FLXAnFRT2C.T2RC is set to 1).

The timer 2 stops when it transits to other state except transition between NORMAL_ACTIVE state and NORMAL_PASSIVE state.

18.2.5.4 FLXAnFRSTPW1 — FlexRay Stop Watch Register 1

The stop watch is activated by the following trigger events.

- Input of a rising edge or falling edge to the FLXA0STPWT pin
- FlexRay 0 interrupt or FlexRay 1 interrupt
- Writing bit FLXAnFRSTPW1.SSWT to 1

With the macrotick counter increment following next to the stop watch activation the actual cycle counter and macrotick values are captured in register FLXAnFRSTPW1 while the slot counter values for channel A and B are captured in register FLXAnFRSTPW2.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 004C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|------------|----|----|----|----|----|-------|-------|------|------|------|------|------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | SMTV[13:0] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | SCCV[5:0] | | | | | — | EINT1 | EINT0 | EETP | SSWT | EDGE | SWMS | ESWT | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.24 FLXAnFRSTPW1 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 to 16 | SMTV[13:0] | Stop Watch Event Occurrence Macrotick Value |
| 15, 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 13 to 8 | SCCV[5:0] | Stop Watch Event Occurrence Cycle Counter Value |
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | EINT1 | FlexRay Interrupt 1 Trigger Enable 0: Stop watch trigger by FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt event triggers stop watch |
| 5 | EINT0 | FlexRay 0 Interrupt Trigger Enable 0: Stop watch trigger by FlexRay 0 interrupt disabled 1: FlexRay interrupt 0 event triggers stop watch |
| 4 | EETP | External Trigger Pin Enable 0: Trigger by the FLXA0STPWT pin is disabled. 1: Trigger by the FLXA0STPWT pin is enabled. |
| 3 | SSWT | Software Stop Watch Trigger 0: Software trigger reset 1: Stop watch activated by software trigger |
| 2 | EDGE | Stop Watch Trigger Edge Select 0: Falling edge 1: Rising edge |

Table 18.24 FLXAnFRSTPW1 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 1 | SWMS | Stop Watch Mode Select 0: Single-shot mode 1: Continuous mode |
| 0 | ESWT | Hardware Stop Watch Trigger Enable 0: Stop watch trigger disabled 1: Stop watch trigger enabled |

(1) FLXAnFRSTPW1.SMTV

Stop Watch Captured Macrotick Value

State of the macrotick counter when the stop watch event occurred.

(2) FLXAnFRSTPW1.SCCV

Stop Watch Captured Cycle Counter Value

State of the cycle counter when the stop watch event occurred.

(3) FLXAnFRSTPW1.EINT1

Enable FlexRay 1 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 1 interrupt when bit FLXAnFRSTPW1.ESWT is 1.

(4) FLXAnFRSTPW1.EINT0

Enable FlexRay 0 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 0 interrupt when bit FLXAnFRSTPW1.ESWT = 1.

(5) FLXAnFRSTPW1.EETP

External Trigger Pin Enable Bit

When bit FLXAnFRSTPW1.ESWT is set to 1, the FLXA0STPWT pin event is treated as a stop watch trigger.

(6) FLXAnFRSTPW1.SSWT

Software Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

Writing 1 in this bit activates the stop watch. This bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

(7) FLXAnFRSTPW1.EDGE

Stop Watch Trigger Edge Select Bit

(8) FLXAnFRSTPW1.SWMS

Stop Watch Mode Select Bit

(9) FLXAnFRSTPW1.ESWT

Enable Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

If enabled, any of an external trigger pin input event, a FlexRay 0 interrupt event, or a FlexRay 1 interrupt event activates the stop watch.

In single-shot mode, this bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

18.2.5.5 FLXAnFRSTPW2 — FlexRay Stop Watch Register 2

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0050_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|-------------|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | SSCVB[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | SSCVA[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.25 FLXAnFRSTPW2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 27 | Reserved | When read, the value after reset is read. |
| 26 to 16 | SSCVB[10:0] | Stop Watch Captured Slot Counter Value Channel B |
| 15 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 0 | SSCVA[10:0] | Stop Watch Captured Slot Counter Value Channel A |

(1) FLXAnFRSTPW2.SSCVB

Stop Watch Captured Slot Counter Value Channel B

State of the slot counter for channel B when the stop watch event occurred.

(2) FLXAnFRSTPW2.SSCVA

Stop Watch Captured Slot Counter Value Channel A

State of the slot counter for channel A when the stop watch event occurred.

18.2.6 CC Control Registers

This section describes the registers provided by the CC (Communication Controller) to allow the Host to control the operation of the CC. The FlexRay protocol specification requires the Host to write application configuration data in CONFIG state only. Please consider that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered from reset. To change POC state from DEFAULT_CONFIG to CONFIG state the Host has to apply CHI command CONFIG. If the Host wants the CC to leave CONFIG state, the Host has to execute the lock release sequence as described in **Section 18.2.3.1, FLXAnFRLCK — FlexRay Lock Register**.

18.2.6.1 FLXAnFRSUCC1 — FlexRay SUC Configuration Register 1

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0080_H

Value after reset: 0C40 1080_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|-----|-----|-----|------|------|------|------|------|-----|------|----------|-----|-----|-----|-----|
| | — | — | — | — | CCHB | CCHA | MTSB | MTSA | HCSE | TSM | WUCS | PTA[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CSA[4:0] | | | | — | TXSY | TXST | PBSY | — | — | — | CMD[3:0] | | | | |
| Value after reset | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 18.26 FLXAnFRSUCC1 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 28 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 27 | CCHB | Connected to Channel B Bit Configures pChannels 0: Not connected to channel B 1: Node connected to channel B (value after reset) |
| 26 | CCHA | Connected to Channel A Bit Configures pChannels 0: Not connected to channel A 1: Node connected to channel A (value after reset) |
| 25 | MTSB | Select Channel B for MTS Transmission Bit 0: Channel B disabled for MTS transmission 1: Channel B selected for MTS transmission |
| 24 | MTSA | Select Channel A for MTS Transmission Bit 0: Channel A disabled for MTS transmission 1: Channel A selected for MTS transmission |
| 23 | HCSE | Halt due to Clock Sync Error Bit Configures pAllowHaltDueToClock 0: CC will enter / remain in NORMAL_PASSIVE 1: CC will enter HALT state |
| 22 | TSM | Transmission Slot Mode Bit Configures pSingleSlotEnabled 0: ALL Slot Mode 1: SINGLE Slot Mode (value after reset) |

Table 18.26 FLXAnFRSUCC1 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 21 | WUCS | Wakeup Channel Select Bit Configures pWakeupChannel 0: Send wakeup pattern on channel A 1: Send wakeup pattern on channel B |
| 20 to 16 | PTA[4:0] | Passive to Active Bit Configures pAllowPassiveToActive |
| 15 to 11 | CSA[4:0] | Cold Start Attempts Bit Configures gColdStartAttempts |
| 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | TXSY | Transmit Sync Frame in Key Slot Bit Configures pKeySlotUsedForSync 0: No sync frame transmission in key slot, node is neither sync nor coldstart node 1: Key slot used to transmit sync frame, node is sync node |
| 8 | TXST | Transmit Startup Frame in Key Slot Bit Configures pKeySlotUsedForStartup 0: No startup frame transmission in key slot, node is non-coldstarter 1: Key slot used to transmit startup frame, node is leading or following coldstarter |
| 7 | PBSY | POC Busy Flag 0: POC not busy, bit FLXAnFRSUCC1.CMD writeable 1: POC is busy, bit FLXAnFRSUCC1.CMD locked |
| 6 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | CMD[3:0] | CHI Command Vector Bit 0000: command_not_accepted 0001: CONFIG 0010: READY 0011: WAKEUP 0100: RUN 0101: ALL_SLOTS 0110: HALT 0111: FREEZE 1000: SEND_MTS 1001: ALLOW_COLDSTART 1010: RESET_STATUS_INDICATORS 1011: MONITOR_MODE 1100: CLEAR_RAMs others: reserved |

(1) FLXAnFRSUCC1.CCHB

Connected to Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Configures whether the node is connected to channel B (pChannels).

(2) FLXAnFRSUCC1.CCHA

Connected to Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Configures whether the node is connected to channel A (pChannels).

(3) FLXAnFRSUCC1.MTSB

Select Channel B for MTS Transmission Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The FLXAnFRSUCC1.MTSB bit may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 18.2.3.1, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1 an MTS symbol will be transmitted on both channels when requested by writing 1000_B to the FLXAnFRSUCC1.CMD[3:0] bits.

The bit selects channel B for MTS symbol transmission.

(4) FLXAnFRSUCC1.MTSA

Select Channel A for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS[5:0] bits are DEFAULT_CONFIG or CONFIG.

Bit FLXAnFRSUCC1.MTSA may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 18.2.3.1, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1 an MTS symbol will be transmitted on both channels when requested by writing 1000_B to the FLXAnFRSUCC1.CMD[3:0] bits.

The bit selects channel A for MTS symbol transmission.

(5) FLXAnFRSUCC1.HCSE

Halt due to Clock Sync Error Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Controls the transition to HALT state due to a clock synchronization error (pAllowHaltDueToClock).

(6) FLXAnFRSUCC1.TSM

Transmission Slot Mode Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] is DEFAULT_CONFIG or CONFIG.

Selects the value after transmission slot mode reset (pSingleSlotEnabled).

In SINGLE slot mode the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 respectively message buffers 0 and 1 depending on bit FLXAnFRMRC.SPLM.

In case FLXAnFRSUCC1.TSM = 1, message buffer 0 respectively message buffers 0,1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode the CC may transmit in all slots.

FLXAnFRSUCC1.TSM is a configuration bit which can only be set / reset by the Host.

The CC changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing 0101_B to bits FLXAnFRSUCC1.CMD[3:0] in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by bits FLXAnFRCCSV.SLM[1:0].

(7) FLXAnFRSUCC1.WUCS

Wakeup Channel Select Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

With this bit the Host selects the channel on which the CC sends the Wakeup pattern (pWakeupChannel).

(8) FLXAnFRSUCC1.PTA

Passive to Active Bit

The user can only write to these bits when the FLXAnFRCCSV.POCS[5:0] bits is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 31 even / odd cycle pairs.

Defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state (pAllowPassiveToActive).

If set to “00000_B” the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

(9) FLXAnFRSUCC1.CSA

Cold Start Attempts Bit

The user can only write to these bits when the FLXAnFRCCSV.POCS[5:0] bits is DEFAULT_CONFIG or CONFIG.

Must be identical in all nodes of a cluster.

Valid values are 2 to 31.

Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node (gColdStartAttempts).

(10) FLXAnFRSUCC1.TXSY

Transmit Sync Frame in Key Slot Bit

The user can only write to these bits when the FLXAnFRCCSV.POCS[5:0] bits is DEFAULT_CONFIG or CONFIG.

Defines whether the key slot is used to transmit sync frames (pKeySlotUsedForSync).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

(11) FLXAnFRSUCC1.TXST

Transmit Startup Frame in Key Slot Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Defines whether the key slot is used to transmit startup frames (pKeySlotUsedForStartup).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

(12) FLXAnFRSUCC1.PBSY

POC Busy Flag

Signals that the POC is busy and cannot accept a command from the Host. Bits FLXAnFRSUCC1.CMD[5:0] are locked against write accesses.

Set to 1 after reset during initialization of internal RAM blocks.

(13) FLXAnFRSUCC1.CMD

CHI Command Vector Bit

The Host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector FLXAnFRSUCC1.CMD[3:0] bits will be reset to 0000_B = command_not_accepted, and flag FLXAnFREIR.CNA will be set to 1.

In general the Host must check FLXAnFRSUCC1.PBSY before writing a new CHI command.

In case the previous CHI command has not yet completed, flag FLXAnFREIR.CCL is set to 1 together with FLXAnFREIR.CNA; the CHI command needs to be repeated.

Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will flag FLXAnFREIR.CNA be set.

Reading bits FLXAnFRSUCC1.CMD[3:0] show whether the last CHI command was accepted. The actual POC state is monitored by bits FLXAnFRCCSV.POCS[5:0].

- command_not_accepted

Bits FLXAnFRSUCC1.CMD[3:0] are reset to 0000_B due to one of the following conditions:

- Illegal command applied by the Host
- Host applied command to leave CONFIG state without preceding config lock key
- Host applied new command while execution of the previous Host command has not completed
- Host writes command_not_accepted

When bits FLXAnFRSUCC1.CMD[3:0] are reset to 0000_B, FLXAnFREIR.CNA is set to 1, and if enabled an interrupt is generated. Commands which are not accepted are not executed.

CONFIG command

Go to POC state CONFIG when called in POC states DEFAULT_CONFIG, or READY. When called in HALT state the CC transits to POC state DEFAULT_CONFIG. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to 0000_B = command_not_accepted.

READY command

Go to POC state READY when called in POC states CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

WAKEUP command

Go to POC state WAKEUP when called in POC state READY. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

RUN command

Go to POC state STARTUP when called in POC state READY. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

ALL_SLOTS command

Leave SINGLE slot mode and go to ALL-SLOTS mode after successful startup / integration at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

HALT command

Set halt request FLXAnFRCCSV.HRQ to 1 and go to POC state HALT at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

FREEZE command

Set the freeze status indicator FLXAnFRCCSV.FSI flag to 1 and go to POC state HALT immediately. Can be called from any state.

SEND_MTS command

Send single MTS symbol during the next following symbol window on the channel configured by the FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB bits, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (the FLXAnFRCCSV.SLM[1:0] bits = “11”). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

ALLOW_COLDSTART command

The command resets the FLXAnFRCCSV.CSI flag to enable the node to become leading coldstarter. When called in states DEFAULT_CONFIG, CONFIG, or HALT, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted. To become leading coldstarter it is also required that both FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY bits are set.

RESET_STATUS_INDICATORS command

Resets status flags FLXAnFRCCSV.FSI, FLXAnFRCCSV.HRQ, FLXAnFRCCSV.CSNI, and FLXAnFRCCSV.CSAI to their values after reset. May be called in POC states READY and STARTUP. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

CLEAR_RAM command

Sets bit FLXAnFRMHDS.CRAME to 1 when called in DEFAULT_CONFIG or CONFIG state. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000_B” = command_not_accepted.

Bit FLXAnFRMHDS.CRAME is also set to 1 when the CC leaves reset. By setting bit FLXAnFRMHDS.CRAME all internal RAM blocks are initialized to zero. During the initialization of the RAMs, bit RSUCC1.PBSY will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAM.

The initialization of the internal message RAM requires 2048 bus clock cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after reset or after assertion of CHI command CLEAR_RAM.

Before asserting CHI command CLEAR_RAM, the Host should make sure that no transfer between Message RAM and IBF / OBF or the TBFRAMs is ongoing and that the data transfer handler has no effect (bits FLXAnFRITS.ITS and FLXAnFROTS.OTS are 0). This command also resets the Message Buffer Status registers FLXAnFRMHDS, FLXAnFRLDTS, FLXAnFRFSR, FLXAnFRMHDF, FLXAnFRTRXQ1 to FLXAnFRTRXQ4, FLXAnFRNDAT1 to FLXAnFRNDAT14, and FLXAnFRMBSC1 to FLXAnFRMBSC4.

CAUTIONS

1. All accepted commands with exception of CLEAR_RAM and SEND_MTS will cause a change of the POC state in the FlexRay domain after at most 8 cycles of the slower of the two clocks “bus clock” and “FlexRay sample clock”, assumed that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register FLXAnFRCCSV will show data that is additionally delayed by synchronization from the FlexRay domain to the bus clock domain. The maximum additional delay is 12 cycles of the slower of the two clocks 'bus clock' and “FlexRay sample clock”.
2. When transfer is stopped by a FREEZE or READY command and then restarted as a leading ColdStart node, the startup frame may not be transmitted in cycle 0. This depends on the internal state of the FlexRay module. Such cases arise when the startup frame is set in a slot 1 to slot 7. This does not occur in a ColdStart after the reset of the microcontroller. Even if the above situation arises, the ColdStart will succeed on the second trial. Repetition prolongs the overall ColdStart process, but the ColdStart will not be obstructed by the earlier situation. To avoid this effect, allocate startup and sync frames to static slot 8 or a slot with a higher number.

Table 18.27 below references the CHI commands from the *FlexRay Protocol Specification (Section 2.1.1.1, Table 2.2)* to the FlexRay CHI command vector FLXAnFRSUCC1.CMD[3:0] bits.

Table 18.27 Reference to CHI Host Command Summary from FlexRay Protocol Specification

| CHI command | Where processed (POC States) | CHI Command Vector CMD |
|--------------------|---|-------------------------------|
| ALL_SLOTS | POC: normal active, POC: normal passive | ALL_SLOTS |
| ALLOW_COLDSTART | All except POC: default config, POC: config, POC: halt | ALLOW_COLDSTART |
| CONFIG | POC: default config, POC: ready | CONFIG |
| CONFIG_COMPLETE | POC: config | Unlock sequence & READY |
| DEFAULT_CONFIG | POC: halt | CONFIG |
| FREEZE | All | FREEZE |
| HALT | POC: normal active, POC: normal passive | HALT |
| READY | All except POC: default config, POC: config, POC: ready, POC: halt | READY |
| RUN | POC: ready | RUN |
| WAKEUP | POC: ready | WAKEUP |

18.2.6.2 FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0084_H

Value after reset: 0100 0504_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|-----|-----|-----|----------|-----|-----|-----|-----|-----|-----------|-----|-----|-----|-----|-----|
| | — | — | — | — | LTN[3:0] | | | — | — | — | LT[20:16] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LT[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.28 FLXAnFRSUCC2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 28 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 27 to 24 | LTN[3:0] | Listen Timeout Noise Bit Configures (gListenNoise - 1) |
| 23 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 0 | LT[20:0] | Listen Timeout Bit Configures pdListenTimeout |

(1) FLXAnFRSUCC2.LTN

Listen Timeout Noise Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The range for gListenNoise is 2 to 16.

FLXAnFRSUCC2.LTN must be configured identical in all nodes of a cluster.

Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout.

CAUTION

The wakeup / startup noise timeout is calculated as follows:

$$\text{pdListenTimeout} \times \text{gListenNoise} = \text{FLXAnFRSUCC2.LT} \times (\text{FLXAnFRSUCC2.LTN} + 1)$$

(2) FLXAnFRSUCC2.LT

Listen Timeout Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The range for pdListenTimeout is 1284 to 1283846 μ T.

Configures wakeup / startup listen time out in μ T.

18.2.6.3 FLXAnFRSUCC3 — FlexRay SUC Configuration Register 3

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0088_H

Value after reset: 0000 0011_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----------|-----|-----|----------|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | WCF[3:0] | | | WCP[3:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.29 FLXAnFRSUCC3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 to 4 | WCF[3:0] | Maximum Without Clock Correction Fatal Bit (transition to HALT state) Configures gMaxWithoutClockCorrectionFatal |
| 3 to 0 | WCP[3:0] | Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state) Configures gMaxWithoutClockCorrectionPassive |

(1) FLXAnFRSUCC3.WCF

Maximum Without Clock Correction Fatal Bit (transition to HALT state)

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15 cycle pairs.

Must be identical in all nodes of a cluster.

Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

CAUTION

The transition to HALT state is prevented if the FLXAnFRSUCC1.HCSE bit is not set.

(2) FLXAnFRSUCC3.WCP

Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state)

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15 cycle pairs.

Must be identical in all nodes of a cluster.

Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state.

18.2.6.4 FLXAnFRNEMC — FlexRay NEM Configuration Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 008C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----------|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | NML[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |

Table 18.30 FLXAnFRNEMC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | NML[3:0] | Network Management Vector Length Bit Configures gNetworkManagementVectorLength |

(1) FLXAnFRNEMC.NML

Network Management Vector Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 12 bytes.

The configured length must be identical in all nodes of a cluster.

These bits configure the length of the NM vector.

18.2.6.5 FLXAnFRPRTC1 — FlexRay PRT Configuration Register 1

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0090_H

Value after reset: 084C 0633_H

| | | | | | | | | | | | | | | | | |
|-------------------|----------|-----|----------|-----|-----|-----------|-----|----------|-----|-----|-----|-----------|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RWP[5:0] | | | | | | — | RXW[8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | BRP[1:0] | | SPP[1:0] | | — | CASM[6:0] | | | | | | TSST[3:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.31 FLXAnFRPRTC1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 26 | RWP[5:0] | Repetitions of Tx Wakeup Pattern Bit Configures pWakeupPattern |
| 25 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 24 to 16 | RXW[8:0] | Wakeup Symbol Receive Window Length Bit Configures gdWakeupSymbolRxWindow |
| 15, 14 | BRP[1:0] | Baud Rate Prescaler Bit Configures gdSampleClockPeriod and pSamplesPerMicrotick 00 = 10 Mbps 01 = 5 Mbps 10 = 2.5 Mbps 11 = 2.5 Mbps |
| 13, 12 | SPP[1:0] | Strobe Point Position Bit Configures Strobe point position 00 = Sample 5 01 = Sample 4 10 = Sample 6 11 = Sample 5 |
| 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 to 4 | CASM[6:0] | Collision Avoidance Symbol Max Bit Configures gdCASRxLowMax |
| 3 to 0 | TSST[3:0] | Transmission Start Sequence Transmitter Bit Configures gdTSSTransmitter |

(1) FLXAnFRPRTC1.RWP

Repetitions of Tx Wakeup Pattern Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63.

Configures the number of repetitions (sequences) of the Tx wakeup symbol.

(2) FLXAnFRPRTC1.RXW

Wakeup Symbol Receive Window Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] DEFAULT_CONFIG or CONFIG.

Valid values are 76 to 301 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the received wakeup pattern.

(3) FLXAnFRPRTC1.BRP

Baud Rate Prescaler Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock set to 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate.

00 = 10 MBit/s

$$\text{gdSampleClockPeriod} = 12.5 \text{ ns} = 1 \times \text{“sample clock”}$$

$$\text{pSamplesPerMicrotick} = 2 \text{ (1 } \mu\text{T} = 25 \text{ ns)}$$

01 = 5 MBit/s

$$\text{gdSampleClockPeriod} = 25 \text{ ns} = 2 \times \text{“sample clock”}$$

$$\text{pSamplesPerMicrotick} = 1 \text{ (1 } \mu\text{T} = 25 \text{ ns)}$$

10, 11 = 2.5 MBit/s

$$\text{gdSampleClockPeriod} = 50 \text{ ns} = 4 \times \text{“sample clock”}$$

$$\text{pSamplesPerMicrotick} = 1 \text{ (1 } \mu\text{T} = 50 \text{ ns)}$$

(4) FLXAnFRPRTC1.SPP

Strobe Point Position Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

These bits specify the sample-counter position at which the Checker Core strobes in response to received bits.

The strobed bit value is determined by sampling at the timing specified by bits FLXAnFRPRTC1.SPP[1:0].

CAUTION

The current revision 2.1 of the FlexRay protocol requires that the FLXAnFRPRTC1.SPP[1:0] bits are 00. The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.

(5) FLXAnFRPRTC1.CASM

Collision Avoidance Symbol Max Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

CASM6 is fixed to 1.

Valid values are 67 to 99 bit times.

Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS).

(6) FLXAnFRPRTC1.TSST

Transmission Start Sequence Transmitter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are is DEFAULT_CONFIG or CONFIG.

Valid values are 3 to 15 bit times.

Must be identical in all nodes of a cluster.

Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = 4 μ T = 100ns @ 10Mbps).

18.2.6.6 FLXAnFRPRTC2 — FlexRay PRT Configuration Register 2

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0094_H

Value after reset: 0F2D 0A0E_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----------|-----|-----|-----|-----|-----|-----|----------|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | TXL[5:0] | | | | | | | TXI[7:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | | RXL[5:0] | | | | | — | | RXI[5:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.32 FLXAnFRPRTC2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 to 24 | TXL[5:0] | Wakeup Symbol Transmit Low Bit Configures gdWakeupSymbolTxLow |
| 23 to 16 | TXI[7:0] | Wakeup Symbol Transmit Idle Bit Configures gdWakeupSymbolTxIdle |
| 15, 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 13 to 8 | RXL[5:0] | Wakeup Symbol Receive Low Bit Configures gdWakeupSymbolRxLow |
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | RXI[5:0] | Wakeup Symbol Rx Idle Bit Configures gdWakeupSymbolRxIdle |

(1) FLXAnFRPRTC2.TXL

Wakeup Symbol Transmit Low Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 15 to 60 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol.

(2) FLXAnFRPRTC2.TXI

Wakeup Symbol Transmit Idle Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 45 to 180 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol.

(3) FLXAnFRPRTC2.RXL

Wakeup Symbol Receive Low Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 55 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol.

(4) FLXAnFRPRTC2.RXI

Wakeup Symbol Rx Idle Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 59 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol.

18.2.6.7 FLXAnFRMHDC — FlexRay MHD Configuration Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0098_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|-----------|-----|-----|-----|-----|-----|-----------|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | SLT[12:0] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | | | | | | | | | SFDL[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.33 FLXAnFRMHDC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 29 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 28 to 16 | SLT[12:0] | Start of Latest Transmit Bit Configures pLatestTx |
| 15 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | SFDL[6:0] | Static Frame Data Length Bit Configures gPayloadLengthStatic |

(1) FLXAnFRMHDC.SLT

Start of Latest Transmit Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7981 minislots.

Configures the maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if FLXAnFRMHDC.SLT is set to zero.

(2) FLXAnFRMHDC.SFDL

Static Frame Data Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 127.

The payload length must be identical in all nodes of a cluster.

Configures the cluster-wide payload length for all frames sent in the static segment in double bytes.

18.2.6.8 FLXAnFRGTUC1 — FlexRay GTU Configuration Register 1

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00A0_H

Value after reset: 0000 0280_H

| | | | | | | | | | | | | | | | | |
|-------------------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | | | | UT[19:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | UT[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.34 FLXAnFRGTUC1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 20 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 19 to 0 | UT[19:0] | Setting of Communication Cycle in Microticks Bit Configures pMicroPerCycle |

(1) FLXAnFRGTUC1.UT

Setting of Communication Cycle in Microticks Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 640 to 640000 μ T.

Configures the duration of the communication cycle in microticks.

18.2.6.9 FLXAnFRGTUC2 — FlexRay GTU Configuration Register 2

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00A4_H

Value after reset: 0002 000A_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | SNM[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | MPC[13:0] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.35 FLXAnFRGTUC2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 20 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 19 to 16 | SNM[3:0] | Sync Node Max Bit |
| 15, 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 13 to 0 | MPC[13:0] | Setting of Communication Cycle in Macro tick Bit Configures gMacroPerCycle |

(1) FLXAnFRGTUC2.SNM

Sync Node Max Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCSDEFAULT_CONFIG or CONFIG.

Valid values are 2 to 15.

Must be identical in all nodes of a cluster.

Maximum number of frames within a cluster with sync frame indicator bit SYN set to 1.

(2) FLXAnFRGTUC2.MPC

Setting of Communication Cycle in Macro tick Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS [5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 16000 MT.

The cycle length must be identical in all nodes of a cluster.

Configures the duration of one communication cycle in macro ticks.

18.2.6.10 FLXAnFRGTUC3 — FlexRay GTU Configuration Register 3

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00A8_H

Value after reset: 0202 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|-----------|-----|-----|-----|-----|-----------|-----|-----|-----------|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | MIOB[6:0] | | | | | | — | | MIOA[6:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | UIOB[7:0] | | | | | | | UIOA[7:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.36 FLXAnFRGTUC3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 30 to 24 | MIOB[6:0] | Macrotick Initial Offset Channel B Bit Configures pMacroInitialOffset[B] |
| 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | MIOA[6:0] | Macrotick Initial Offset Channel A Bit Configures pMacroInitialOffset[A] |
| 15 to 8 | UIOB[7:0] | Microtick Initial Offset Channel B Bit Configures pMicroInitialOffset[B] |
| 7 to 0 | UIOA[7:0] | Microtick Initial Offset Channel A Bit Configures pMicroInitialOffset[A] |

(1) FLXAnFRGTUC3.MIOB

Macrotick Initial Offset Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster.

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration.

(2) FLXAnFRGTUC3.MIOA

Macrotock Initial Offset Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster.

Configures the number of macroticks between the static slot boundary and the subsequent macrotock boundary of the secondary time reference point based on the nominal macrotock duration.

(3) FLXAnFRGTUC3.UIOB

Microtock Initial Offset Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCSDEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 μ T.

Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotock boundary of the secondary time reference point. The parameter depends on pDelayCompensation [B] and therefore has to be set for each channel independently.

(4) FLXAnFRGTUC3.UIOA

Microtock Initial Offset Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS [5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 μ T.

Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotock boundary of the secondary time reference point. The parameter depends on pDelayCompensation [A] and therefore has to be set for each channel independently.

18.2.6.11 FLXAnFRGTUC4 — FlexRay GTU Configuration Register 4

For details about configuration of FLXAnFRGTUC4.NIT and FLXAnFRGTUC4.OCS see **Section 18.3.2.5, Configuration of NIT Start and Offset Correction Start.**

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00AC_H

Value after reset: 0008 0007_H

| | | | | | | | | | | | | | | | | | |
|-------------------|----|----|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| | — | — | OCS[13:0] | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | — | — | NIT[13:0] | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

Table 18.37 FLXAnFRGTUC4 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 to 16 | OCS[13:0] | Offset Correction Start Bit Configures (gOffsetCorrectionStart - 1) |
| 15, 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 13 to 0 | NIT[13:0] | Network Idle Time Start Bit Configures (gMacroPerCycle -gdNIT - 1) |

(1) FLXAnFRGTUC4.OCS

Offset Correction Start Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 8 to 15998 MT.

For cluster consisting of E-Ray implementations only, it is sufficient to program
FLXAnFRGTUC4.OCS = FLXAnFRGTUC4.NIT + 1.

Must be identical in all nodes of a cluster.

Determines the start of the offset correction within the NIT phase, calculated from start of cycle.

(2) FLXAnFRGTUC4.NIT

Network Idle Time Start Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 7 to 15997 MT.

Must be identical in all nodes of a cluster.

Configures the starting point of the Network Idle Time NIT at the end of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of NIT is recognized if $\text{Macrotick} = \text{gMacroPerCycle} - \text{gdNIT} - 1$ and the increment pulse of Macrotick is set.

18.2.6.12 FLXAnFRGTUC5 — FlexRay GTU Configuration Register 5

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00B0_H

Value after reset: 0E00 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|-----|-----|-----|-----|-----|-----|----------|-----|-----|----------|-----|-----|-----|-----|-----|
| | DEC[7:0] | | | | | | | — | — | — | CDD[4:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DCB[7:0] | | | | | | | DCA[7:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.38 FLXAnFRGTUC5 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 24 | DEC[7:0] | Decoding Correction Bit Configures pDecodingCorrection |
| 23 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 16 | CDD[4:0] | Cluster Drift Damping Bit Configures pClusterDriftDamping |
| 15 to 8 | DCB[7:0] | Delay Compensation Channel B Bit Configures pDelayCompensation[B] |
| 7 to 0 | DCA[7:0] | Delay Compensation Channel A Bit Configures pDelayCompensation[A] |

(1) FLXAnFRGTUC5.DEC

Decoding Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 143 μ T.

Configures the decoding correction value in microticks used to determine the primary time reference point.

(2) FLXAnFRGTUC5.CDD

Cluster Drift Damping Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 20 μ T.

Configures the cluster drift damping value in microticks used in clock synchronization to minimize accumulation of rounding errors.

(3) FLXAnFRGTUC5.DCB

Delay Compensation Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 μ T.

Used to compensate for reception delays on channel B. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 μ s. In practice, the minimum of the propagation delays of all sync nodes should be applied.

(4) FLXAnFRGTUC5.DCA

Delay Compensation Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 μ T.

Used to compensate for reception delays on channel A. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 μ s. In practice, the minimum of the propagation delays of all sync nodes should be applied.

18.2.6.13 FLXAnFRGTUC6 — FlexRay GTU Configuration Register 6

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00B4_H

Value after reset: 0002 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | MOD[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | ASR[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.39 FLXAnFRGTUC6 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 to 16 | MOD[10:0] | Maximum Oscillator Drift Bit Configures pdMaxDrift |
| 15 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 to 0 | ASR[10:0] | Accepted Startup Range Bit Configures pdAcceptedStartupRange |

(1) FLXAnFRGTUC6.MOD

Maximum Oscillator Drift Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 μ T.

Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in μ T.

(2) FLXAnFRGTUC6.ASR

Accepted Startup Range Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 1875 μ T.

Number of microticks constituting the expanded range of measured deviation for startup frames during integration.

18.2.6.14 FLXAnFRGTUC7 — FlexRay GTU Configuration Register 7

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00B8_H

Value after reset: 0002 0004_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | NSS[9:0] | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | | | | | | SSL[9:0] | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.40 FLXAnFRGTUC7 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 26 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 25 to 16 | NSS[9:0] | Number of Static Slots Bit Configures gNumberOfStaticSlots |
| 15 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 to 0 | SSL[9:0] | Static Slot Length Bit Configures gdStaticSlot |

(1) FLXAnFRGTUC7.NSS

Number of Static Slots Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1023.

The number of static slots must be identical in all nodes of a cluster.

Configures the number of static slots in a cycle.

(2) FLXAnFRGTUC7.SSL

Static Slot Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 4 to 659 MT.

The static slot length must be identical in all nodes of a cluster.

Configures the length of a static slot in macroticks.

18.2.6.15 FLXAnFRGTUC8 — FlexRay GTU Configuration Register 8

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00BC_H

Value after reset: 0000 0002_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|-----------|-----|-----|-----|-----|-----|-----|----------|-----|-----|-----|-----|-----|
| | — | — | — | NMS[12:0] | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | MSL[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.41 FLXAnFRGTUC8 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 29 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 28 to 16 | NMS[12:0] | Number of Minislots Bit Configures gNumberOfMinislots |
| 15 to 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | MSL[5:0] | Minislot Length Bit Configures gdMinislot |

(1) FLXAnFRGTUC8.NMS

Number of Minislots Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7986.

The number of minislots must be identical in all nodes of a cluster.

Configures the number of minislots within the dynamic segment of a cycle.

(2) FLXAnFRGTUC8.MSL

Minislot Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63 MT.

The minislot length must be identical in all nodes of a cluster.

Configures the length of a minislot in macroticks.

18.2.6.16 FLXAnFRGTUC9 — FlexRay GTU Configuration Register 9

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00C0_H

Value after reset: 0000 0101_H

| | | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|-----|-----------|-----|-----|-----|----|----|-----|-----|----------|-----|----------|-----|--|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| | — | | | | | | | | | | | | | | DSI[1:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | — | | — | | MAPO[4:0] | | | | — | | — | | APO[5:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | |

Table 18.42 FLXAnFRGTUC9 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17, 16 | DSI[1:0] | Dynamic Slot Idle Phase Bit Configures gdDynamicSlotIdlePhase |
| 15 to 13 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 12 to 8 | MAPO[4:0] | Minislot Action Point Offset Bit Configures gdMinislotActionPointOffset |
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | APO[5:0] | Action Point Offset Bit Configures gdActionPointOffset |

(1) FLXAnFRGTUC9.DSI

Dynamic Slot Idle Phase Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 2 Minislot.

Must be identical in all nodes of a cluster.

Configures the duration of the dynamic slot idle phase in the number of minislots. The duration has to be greater or equal than the idle detection time.

(2) FLXAnFRGTUC9.MAPO

Minislot Action Point Offset Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 31 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within the minislots of the dynamic segment.

(3) FLXAnFRGTUC9.APO

Action Point Offset Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 63 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within static slots and symbol window.

18.2.6.17 FLXAnFRGTUC10 — FlexRay GTU Configuration Register 10

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00C4_H

Value after reset: 0002 0005_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|-----------|-----|-----|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | MRC[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | MOC[13:0] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.43 FLXAnFRGTUC10 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 to 16 | MRC[10:0] | Maximum Rate Correction Bit Configures pRateCorrectionOut |
| 15, 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 13 to 0 | MOC[13:0] | Maximum Offset Correction Bit Configures pOffsetCorrectionOut |

(1) FLXAnFRGTUC10.MRC

Maximum Rate Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 μ T.

Holds the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks only the internal rate correction value against the maximum rate correction value (absolute value).

(2) FLXAnFRGTUC10.MOC

Maximum Offset Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 5 to 15266 μ T.

Holds the maximum permitted offset correction value (absolute value) to be applied by the internal clock synchronization algorithm (absolute value). The CC checks only the internal offset correction value against the maximum offset correction value.

18.2.6.18 FLXAnFRGTUC11 — FlexRay GTU Configuration Register 11

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 00C8_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----------|-----------|-----|----|----|----|----|----|----------|-----------|-----|
| | — | — | — | — | — | ERC[2:0] | | | — | — | — | — | — | EOC[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | ERCC[1:0] | | — | — | — | — | — | — | EOCC[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R/W | R/W |

Table 18.44 FLXAnFRGTUC11 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 to 24 | ERC[2:0] | External Rate Correction Bit Configures pExternRateCorrection |
| 23 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 to 16 | EOC[2:0] | External Offset Correction Bit Configures pExternOffsetCorrection |
| 15 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9, 8 | ERCC[1:0] | External Rate Correction Control Bit Configures vExternRateControl 00: External rate correction is prohibited. 01: External rate correction is prohibited. 10: Subtract 11: Add |
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | EOCC[1:0] | External Offset Correction Control Bit Configures vExternOffsetControl 00: External offset correction is prohibited. 01: External offset correction is prohibited. 10: Subtract 11: Add |

(1) FLXAnFRGTUC11.ERC

External Rate Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 μ T.

Holds the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated rate correction value. The value is applied during NIT.

(2) FLXAnFRGTUC11.EOC

External Offset Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 μ T.

Holds the external offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated offset correction value. The value is applied during NIT.

(3) FLXAnFRGTUC11.ERCC

External Rate Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

Writing the following values enables the external rate correction.

00 = External rate correction is prohibited.

01 = External rate correction is prohibited.

10 = Subtract

External rate correction value subtracted from calculated rate correction value

11 = Add

External rate correction value added to calculated rate correction value

(4) FLXAnFRGTUC11.EOCC

External Offset Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

Writing the following values enables the external offset correction.

00 = External offset correction is prohibited.

01 = External offset correction is prohibited.

10 = Subtract

External offset correction value subtracted from calculated offset correction value

11 = Add

External offset correction value added to calculated offset correction value

18.2.7 CC Status Registers

During 8/16-bit accesses to status variables coded with more than 8/16-bit, the variable might be updated by the CC between two accesses (non-atomic read accesses).

18.2.7.1 FLXAnFRCCSV — FlexRay CC Status Vector Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0100_H

Value after reset: 0010 4000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|-----|----------|------|----|----|----------|----------|-----|-----|-----------|----------|----|----|----|----|
| | — | — | PSL[5:0] | | | | | RCA[4:0] | | | | WSV[2:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | CSI | CSAI | CSNI | — | — | SLM[1:0] | | HRQ | FSI | POCS[5:0] | | | | | |
| Value after reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.45 FLXAnFRCCSV Register Contents (1/2)

| Bit | Symbol | Function |
|----------|----------|--|
| 31, 30 | Reserved | When read, the value after reset is read. |
| 29 to 24 | PSL[5:0] | POC Status Log Flag Status of FLXAnFRCCSV.POCS immediately before entering HALT state. |
| 23 to 19 | RCA[4:0] | Remaining Coldstart Attempts Flag Indicates vRemainingColdstartAttempts |
| 18 to 16 | WSV[2:0] | Wakeup Status Flag Indicates vPOC!WakeupStatus 000: UNDEFINED 001: RECEIVED_HEADER 010: RECEIVED_WUP 011: COLLISION_HEADER 100: COLLISION_WUP 101: COLLISION_UNKNOWN 110: TRANSMITTED 111: Reserved |
| 15 | Reserved | When read, the value after reset is read. |
| 14 | CSI | Cold Start Inhibit Flag Indicates vColdStartInhibit 0: Cold starting of node enabled 1: Cold starting of node disabled |
| 13 | CSAI | Coldstart Abort Indicator Flag |
| 12 | CSNI | Coldstart Noise Indicator Flag Indicates vPOC!ColdstartNoise |
| 11, 10 | Reserved | When read, the value after reset is read. |
| 9, 8 | SLM[1:0] | Slot Mode Flag Indicates vPOC!SlotMode 00: SINGLE 01: reserved 10: ALL_PENDING 11: ALL |

Table 18.45 FLXAnFRCCSV Register Contents (2/2)

| Bit | Symbol | Function |
|--------|-----------|---|
| 7 | HRQ | Halt Request Flag Indicates vPOC!CHI!HaltRequest |
| 6 | FSI | Freeze Status Indicator Flag Indicates vPOC!Freeze |
| 5 to 0 | POCS[5:0] | Protocol Operation Control Status Flag |

(1) FLXAnFRCCSV.PSL

POC Status Log Flag

Set the value of bits FLXAnFRCCSV.POCS[5:0] immediately before entering HALT state.

Set to HALT when FREEZE command is applied during HALT state and FLXAnFRCCSV.FSI is not already set i.e. the HALT state was not reached by FREEZE command.

Reset to “000000_B” when leaving HALT state.

(2) FLXAnFRCCSV.RCA

Remaining Coldstart Attempts Flag

Indicates the number of remaining coldstart attempts (vRemainingColdstartAttempts).

The value after a reset during CONFIG and DEFAULT_CONFIG state is also bits FLXAnFRSUCC1.CSA[4:0].

The RUN command resets this counter to the maximum number of coldstart attempts as configured by the FLXAnFRSUCC1.CSA [4:0] bits.

(3) FLXAnFRCCSV.WSV

Wakeup Status Flag

Indicates the status of the current wakeup attempt (vPOC!WakeupStatus).

Reset to 0 when entering Wakeup state, by CHI command RESET_STATUS_INDICATORS, or by transition from DEFAULT_CONFIG to CONFIG state

000_B = UNDEFINED

Wakeup not yet executed by the CC.

001_B = RECEIVED_HEADER

Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state.

010_B = RECEIVED_WUP

Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state.

011_B = COLLISION_HEADER

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel.

100_B = COLLISION_WUP

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel.

101_B = COLLISION_UNKNOWN

Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header.

110_B = TRANSMITTED

Set when the CC has successfully completed the transmission of the wakeup pattern.

111_B = reserved

(4) FLXAnFRCCSV.CSI

Cold Start Inhibit Flag

Indicates that the node is disabled from cold starting (vColdStartInhibit).

The flag is set to 1 whenever the POC enters READY state due to CHI command READY.

The flag has to be reset under control of the Host by CHI command ALLOW_COLDSTART (bits FLXAnFRSUCC1.CMD[3:0] are 1001_B).

(5) FLXAnFRCCSV.CSAI

Coldstart Abort Indicator Flag

Indicates that a coldstart attempt was aborted.

Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(6) FLXAnFRCCSV.CSNI

Coldstart Noise Indicator Flag

Indicates that the cold start procedure occurred under noisy conditions (vPOC!ColdstartNoise).

Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(7) FLXAnFRCCSV.SLM

Slot Mode Flag

Indicates the actual slot mode of the POC (vPOC!SlotMode) in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE.

Default value is SINGLE. Changes to ALL, depending on FLXAnFRSUCC1.TSM.

In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL.

Set FLXAnFRSUCC1.TSM to SINGLE except for NORMAL_ACTIVE or NORMAL_PASSIVE.

(8) FLXAnFRCCSV.HRQ

Halt Request Flag

Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle (vPOC!CHIHaltRequest).

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(9) FLXAnFRCCSV.FSI

Freeze Status Indicator Flag

Indicates that the POC has entered the HALT state due to CHI command, FREEZE (bits FLXAnFRSUCC1.CMD[3:0] are 0111_B) or due to an error condition requiring an immediate POC halt (vPOC!Freeze).

Reset by transition from HALT to DEFAULT_CONFIG state.

(10) FLXAnFRCCSV.POCS

Protocol Operation Control Status Flag

Indicates the actual state of operation of the CC Protocol Operation Control

00 0000_B = DEFAULT_CONFIG state

00 0001_B = READY state

00 0010_B = NORMAL_ACTIVE state

00 0011_B = NORMAL_PASSIVE state

00 0100_B = HALT state

00 1111_B = CONFIG state

Indicates the actual state of operation of the POC in the wakeup path

01 0000_B = WAKEUP_STANDBY state

01 0001_B = WAKEUP_LISTEN state

01 0010_B = WAKEUP_SEND state

01 0011_B = WAKEUP_DETECT state

Indicates the actual state of operation of the POC in the startup path

10 0000_B = STARTUP_PREPARE state

10 0001_B = COLDSTART_LISTEN state

10 0010_B = COLDSTART_COLLISION_RESOLUTION state

10 0011_B = COLDSTART_CONSISTENCY_CHECK state

10 0100_B = COLDSTART_GAP state

10 0101_B = COLDSTART_JOIN State

10 0110_B = INTEGRATION_COLDSTART_CHECK state

10 0111_B = INTEGRATION_LISTEN state

10 1000_B = INTEGRATION_CONSISTENCY_CHECK state

10 1001_B = INITIALIZE_SCHEDULE state

10 1010_B = ABORT_STARTUP state

10 1011_B = STARTUP_SUCCESS state

others = reserved

18.2.7.2 FLXAnFRCCEV — FlexRay CC Error Vector Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0104_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|-----------|----|----|----|-----------|----|----|----|-----------|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | PTAC[4:0] | | | | ERRM[1:0] | | — | — | CCFC[3:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.46 FLXAnFRCCEV Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 13 | Reserved | When read, the value after reset is read. |
| 12 to 8 | PTAC[4:0] | Passive to Active Count Flag Indicates vAllowPassiveToActive |
| 7, 6 | ERRM[1:0] | Error Mode Flag Indicates vPOC!ErrorMode 00: ACTIVE 01: PASSIVE 10: COMM_HALT 11: reserved |
| 5, 4 | Reserved | When read, the value after reset is read. |
| 3 to 0 | CCFC[3:0] | Clock Correction Failed Counter Indicates vClockCorrectionFailed |

(1) FLXAnFRCCEV.PTAC

Passive to Active Count Flag

Indicates the number of consecutive even / odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when this bit equals to the value of bits FLXAnFRSUCC1.PTA[4:0] –1.

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(2) FLXAnFRCCEV.ERRM

Error Mode Flag

Indicates the actual error mode of the POC (vPOC!ErrorMode).

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(3) FLXAnFRCCEV.CCFC

Clock Correction Failed Counter

Indicates the clock correction failed counter of the POC (vClockCorrectionFailed).

The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error are active.

The Clock Correction Failed Counter is reset to 0 at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active.

The Clock Correction Failed Counter stops at 15.

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

18.2.7.3 FLXAnFRSCV — FlexRay Slot Counter Value Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0110_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | SCCB[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | SCCA[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.47 FLXAnFRSCV Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 27 | Reserved | When read, the value after reset is read. |
| 26 to 16 | SCCB[10:0] | Slot Counter Channel B Indicates vSlotCounter[B] |
| 15 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 0 | SCCA[10:0] | Slot Counter Channel A Indicates vSlotCounter[A] |

(1) FLXAnFRSCV.SCCB

Slot Counter Channel B

Current slot counter value on channel B (vSlotCounter[B]). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSCV.SCCA

Slot Counter Channel A

Current slot counter value on channel A (vSlotCounter[A]). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

18.2.7.4 FLXAnFRMTCCV — FlexRay Macrotick and Cycle Counter Value Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0114_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|-----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | CCV[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | MTV[13:0] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.48 FLXAnFRMTCCV Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 22 | Reserved | When read, the value after reset is read. |
| 21 to 16 | CCV[5:0] | Cycle Counter Value Indicates vCycleCounter |
| 15, 14 | Reserved | When read, the value after reset is read. |
| 13 to 0 | MTV[13:0] | Macrotick Value Indicates vMacrotick |

(1) FLXAnFRMTCCV.CCV

Cycle Counter Value

Current cycle counter value (vCycleCounter). The value is incremented by the CC at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRMTCCV.MTV

Macrotick Value

Current macrotick value (vMacrotick). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

18.2.7.5 FLXAnFRRCV — FlexRay Rate Correction Value Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0118_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|-----------|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | RCV[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.49 FLXAnFRRCV Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 12 | Reserved | When read, the value after reset is read. |
| 11 to 0 | RCV[11:0] | Rate Correction Value Flag Indicates vRateCorrection |

(1) FLXAnFRRCV.RCV

Rate Correction Value Flag

Indicates internal rate correction value (vRateCorrection/ two's complement) before limitation. If the value of this bit exceeds the limits defined by bits FLXAnFRGTUC10.MRC[10:0], flag FLXAnFRSFS.RCLR is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

CAUTION

The external rate correction value is added to the limited rate correction value.

18.2.7.6 FLXAnFROCV — FlexRay Offset Correction Value Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 011C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | OCV[18:16] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OCV[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.50 FLXAnFROCV Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 19 | Reserved | When read, the value after reset is read. |
| 18 to 0 | OCV[18:0] | Offset Correction Value Flag Indicates vOffsetCorrection |

(1) FLXAnFROCV.OCV

Offset Correction Value Flag

Indicates offset correction value (vOffsetCorrection/ two's complement) before limitation. If the value of this bit exceeds the limits defined by bits FLXAnFRGTUC10.MOC[10:0], flag FLXAnFRSFS.OCLR is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

CAUTION

The external offset correction value is added to the limited offset correction value.

18.2.7.7 FLXAnFRSFS — FlexRay Sync Frame Status Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0120_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|----|----|----|-----------|----|----|----|-----------|----|----|----|-----------|------|------|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | RCLR | MRCS | OCLR | MOCS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | VSBO[3:0] | | | | VSBE[3:0] | | | | VSAO[3:0] | | | | VSAE[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.51 FLXAnFRSFS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 20 | Reserved | When read, the value after reset is read. |
| 19 | RCLR | Rate Correction Limit Reached Flag 0: Rate correction below limit 1: Rate correction limit reached |
| 18 | MRCS | Missing Rate Correction Signal Flag 0: Rate correction signal valid 1: Missing rate correction signal |
| 17 | OCLR | Offset Correction Limit Reached Flag 0: Offset correction below limit 1: Offset correction limit reached |
| 16 | MOCS | Missing Offset Correction Signal Flag 0: Offset correction signal valid 1: Missing offset correction signal |
| 15 to 12 | VSBO[3:0] | Valid Sync Frames Channel B, odd communication cycle |
| 11 to 8 | VSBE[3:0] | Valid Sync Frames Channel B, even communication cycle |
| 7 to 4 | VSAO[3:0] | Valid Sync Frames Channel A, odd communication cycle |
| 3 to 0 | VSAE[3:0] | Valid Sync Frames Channel A, even communication cycle |

(1) FLXAnFRSFS.RCLR

Rate Correction Limit Reached Flag

The Rate Correction Limit Reached flag signals to the Host, that the rate correction value has exceeded its limit as defined by bits FLXAnFRGTUC10.MRC[10:0]. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(2) FLXAnFRSFS.MRCS

Missing Rate Correction Signal Flag

The Missing Rate Correction flag signals to the Host, that no rate correction calculation can be performed because no pairs of even / odd sync frames were received. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(3) FLXAnFRSFS.OCLR

Offset Correction Limit Reached Flag

The Offset Correction Limit Reached flag signals to the Host, that the offset correction value has exceeded its limit as defined by bits FLXAnFRGTUC10.MOC[10:0]. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(4) FLXAnFRSFS.MOCS

Missing Offset Correction Signal Flag

The Missing Offset Correction flag signals to the Host, that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(5) FLXAnFRSFS.VSBO

Valid Sync Frames Channel B, odd communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHB is 1.

Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(6) FLXAnFRSFS.VSBE

Valid Sync Frames Channel B, even communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHB is 1.

Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(7) FLXAnFRSFS.VSAO

Valid Sync Frames Channel A, odd communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHA is 1.

Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(8) FLXAnFRSFS.VSAE

Valid Sync Frames Channel A, even communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHA is 1.

Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY, the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

18.2.7.8 FLXAnFRSWNIT — FlexRay Symbol Window and NIT Status Register

Symbol window related status information is updated by the CC at the end of the symbol window for each channel. NIT related status information is updated by the CC at the end of the NIT for each channel.

During startup the status data is not updated.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0124_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|------|------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | SBNB | SENB | SBNA | SENA | MTSB | MTSA | TCSB | SBSB | SESB | TCSA | SBSA | SESA |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.52 FLXAnFRSWNIT Register Contents (1/2)

| Bit | Symbol | Function |
|----------|----------|--|
| 31 to 12 | Reserved | When read, the value after reset is read. |
| 11 | SBNB | Slot Boundary Violation during NIT Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel B |
| 10 | SENB | Syntax Error during NIT Channel B Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel B |
| 9 | SBNA | Slot Boundary Violation during NIT Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel A |
| 8 | SENA | Syntax Error during NIT Channel A Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel A |
| 7 | MTSB | MTS Received on Channel B Flag 0: No MTS symbol received on channel B 1: MTS symbol received on channel B |
| 6 | MTSA | MTS Received on Channel A Flag 0: No MTS symbol received on channel A 1: MTS symbol received on channel A |
| 5 | TCSB | Transmission Conflict in Symbol Window Channel B Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel B |
| 4 | SBSB | Slot Boundary Violation in Symbol Window Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel B |
| 3 | SESB | Syntax Error in Symbol Window Channel B Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel B |

Table 18.52 FLXAnFRSWNIT Register Contents (2/2)

| Bit | Symbol | Function |
|-----|--------|--|
| 2 | TCSA | Transmission Conflict in Symbol Window Channel A Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel A |
| 1 | SBSA | Slot Boundary Violation in Symbol Window Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel A |
| 0 | SESA | Syntax Error in Symbol Window Channel A Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel A |

(1) FLXAnFRSWNIT.SBNB

Indicates a Slot Boundary Violation during NIT Channel B Flag (vSS!BViolationB).

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSWNIT.SENB

Indicates a Syntax Error during NIT Channel B Flag (vSS!SyntaxErrorB).

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRSWNIT.SBNA

Indicates a Slot Boundary Violation during NIT Channel A Flag (vSS!BViolationA).

Reset when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRSWNIT.SENA

Indicates a Syntax Error during NIT Channel A Flag (vSS!SyntaxErrorA).

Reset when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRSWNIT.MTSB

Indicates a MTS Received on Channel B Flag (vSS!ValidMTSB).

Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to 1, also bit FLXAnFRSIR.MTSB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRSWNIT.MTSA

Indicates a MTS Received on Channel A Flag (vSS!ValidMTSA).

Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to 1, also bit FLXAnFRSIR.MTSA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRSWNIT.TCSB

Indicates a Transmission Conflict in Symbol Window Channel B Flag (vSS!TxConflictB).

Reset when leaving CONFIG state or when entering STARTUP state.

(8) FLXAnFRSWNIT.SBSB

Indicates a Slot Boundary Violation in Symbol Window Channel B Flag (vSS!BViolationB).

Reset when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRSWNIT.SESB

Indicates a Syntax Error in Symbol Window Channel B Flag (vSS!SyntaxErrorB).

Reset when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRSWNIT.TCSA

Indicates a Transmission Conflict in Symbol Window Channel A Flag (vSS!TxConflictA).

Reset when leaving CONFIG state or when entering STARTUP state.

(11) FLXAnFRSWNIT.SBSA

Indicates a Slot Boundary Violation in Symbol Window Channel A Flag (vSS!BViolationA).

Reset when leaving CONFIG state or when entering STARTUP state.

(12) FLXAnFRSWNIT.SESA

Indicates a Syntax Error in Symbol Window Channel A Flag (vSS!SyntaxErrorA).

Reset when leaving CONFIG state or when entering STARTUP state.

18.2.7.9 FLXAnFRACS — FlexRay Aggregated Channel Status Register

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception.

The aggregated channel status also includes status data from the symbol window and the network idle time.

The status data is updated (set) after each slot and aggregated until it is reset by the Host.

During startup the status data is not updated.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0128_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|------|-----|------|------|------|----|----|----|------|-----|------|------|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | SBVB | CIB | CEDB | SEDB | VFRB | — | — | — | SBVA | CIA | CEDA | SEDA | VFRA |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 18.53 FLXAnFRACS Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 13 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 12 | SBVB | Slot Boundary Violation on Channel B Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel B |
| 11 | CIB | Communication Indicator Channel B Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel B in slots containing any additional communication |
| 10 | CEDB | Content Error Detected on Channel B Flag 0: No frame with content error received 1: Frame(s) with content error received on channel B |
| 9 | SEDB | Syntax Error Detected on Channel B Flag 0: No syntax error observed 1: Syntax error(s) observed on channel B |
| 8 | VFRB | Valid Frame Received on Channel B Flag 0: No valid frame received 1: Valid frame(s) received on channel B |
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | SBVA | Slot Boundary Violation on Channel A Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel A |

Table 18.53 FLXAnFRACS Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 3 | CIA | Communication Indicator Channel A Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel A in slots containing any additional communication |
| 2 | CEDA | Content Error Detected on Channel A Flag 0: No frame with content error received 1: Frame(s) with content error received on channel A |
| 1 | SEDA | Syntax Error Detected on Channel A Flag 0: No syntax error observed 1: Syntax error(s) observed on channel A |
| 0 | VFRA | Valid Frame Received on Channel A Flag 0: No valid frame received 1: Valid frame(s) received on channel A |

(1) FLXAnFRACS.SBVB

Slot Boundary Violation on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRACS.CIB

Communication Indicator Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had When any combination of either syntax error OR content error OR slot boundary violation.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

The FLXAnFRACS.CIB flag is also set when the slot boundary at the end of the slot is reached during the channel idle recognition phase while there is only a single frame in the slot.

(3) FLXAnFRACS.CEDB

Content Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRACS.SEDB

Syntax Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel B.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRACS.VFRB

Valid Frame Received on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel B in any static or dynamic slot during the observation period.

Reset when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRACS.SBVA

Slot Boundary Violation on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from 0 to 1, bit FLXAnFREIR.EDA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRACS.CIA

Communication Indicator Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

When this flag changes from 0 to 1, the FLXAnFREIR.EDA bit is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

The FLXAnFRACS.CIB flag is also set when the slot boundary at the end of the slot is reached during the channel idle recognition phase while there is only a single frame in the slot.

(8) FLXAnFRACS.CEDA

Content Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRACS.SEDA

Syntax Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel A.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRACS.VFRA

Valid Frame Received on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel A in any static or dynamic slot during the observation period.

Reset when leaving CONFIG state or when entering STARTUP state.

18.2.7.10 FLXAnFRESIDm — FlexRay Even Sync ID Register m (m = 1 to 15)

Registers FLXAnFRESID1 to FLXAnFRESID15 hold the frame IDs of the sync frames received in even communication cycles used for clock synchronisation up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFRESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FLXAnFRESID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFRESID1.RXEA, FLXAnFRESID1.RXEB are set. The value is updated during the NIT of each even communication cycle.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0130_H to <FLXAn_base> + 0168_H (<FLXAn_base> + 0130_H + (n-1) × 4)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------|------|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXEB | RXEA | — | — | — | — | EID[9:0] | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.54 FLXAnFRESIDn (n=1 to 15) Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. |
| 15 | RXEB | Received / Configured Even Sync ID on Channel B Flag 0: No sync frame received on channel B / node not configured to transmit sync frames 1: Sync frame received on channel B / node configured to transmit sync frames |
| 14 | RXEA | Received / Configured Even Sync ID on Channel A Flag 0: No sync frame received on channel A / node not configured to transmit sync frames 1: Sync frame received on channel A / node configured to transmit sync frames |
| 13 to 10 | Reserved | When read, the value after reset is read. |
| 9 to 0 | EID[9:0] | Even Sync ID Flag (vsSyncIDListA,B even) |

(1) FLXAnFRESIDn.RXEB

Received / Configured Even Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = bits FLXAnFRESID1.EID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRESIDn.RXEA

Received / Configured Even Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = bits FLXAnFRESID1.EID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRESIDn.EID

Even Sync ID Flag (vsSyncIDListA,B even)

Sync frame ID even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

18.2.7.11 FLXAnFROSIDm — FlexRay Odd Sync ID Register m (m = 1 to 15)

Registers FLXAnFROSID1 to FLXAnFROSID15 hold the frame IDs of the sync frames received in odd communication cycles used for clock synchronisation up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFROSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FLXAnFROSID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFROSID1.RXOA, FLXAnFROSID1.RXOB are set. The value is updated during the NIT of each odd communication cycle.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0170_H to <FLXAn_base> + 01A8_H (<FLXAn_base> + 0170_H + (n-1) × 4)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------|------|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXOB | RXOA | — | — | — | — | OID[9:0] | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.55 FLXAnFROSIDn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. |
| 15 | RXOB | Received / Configured Odd Sync ID on Channel B Flag 0: No sync frame received on channel B / node not configured to transmit sync frames 1: Sync frame received on channel B / node configured to transmit sync frames |
| 14 | RXOA | Received / Configured Odd Sync ID on Channel A Flag 0: No sync frame received on channel A / node not configured to transmit sync frames 1: Sync frame received on channel A / node configured to transmit sync frames |
| 13 to 10 | Reserved | When read, the value after reset is read. |
| 9 to 0 | OID[9:0] | Odd Sync ID Flag (vsSyncIDListA,B odd) |

(1) FLXAnFROSIDn.RXOB

Received / Configured Odd Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = bits FLXAnFROSID1.OID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFROSIDn.RXOA

Received / Configured Odd Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = bits FLXAnFROSID1.OID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFROSIDn.OID

Odd Sync ID Flag (vsSyncIDListA,B odd)

Sync frame ID odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

18.2.7.12 FLXAnFRNMVm — FlexRay Network Management Vector Register m (m = 1 to 3)

The three network management registers hold the accrued NM vector (see **Section 18.3.7, Network Management**).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

FLXAnFRNMVm-bytes exceeding the configured NM vector length are not valid.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 01B0_H to <FLXAn_base> + 01B8_H (<FLXAn_base> + 01B0_H + (m-1) × 4)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | NM[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | NM[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.56 FLXAnFRNMVn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|-----------|
| 31 to 0 | NM[31:0] | NM Vector |

(1) FLXAnFRNMVm.NM

NM Vector

The three network management vector registers hold the accrued NM vector (configurable 0 to 12 bytes). The NM vector to be held is generated by bit-wise logic OR for each NM vector received on each channel (valid static frames with PPI = 1) (see **Section 18.3.7, Network Management**).

FLXAnFRNMVm-bytes exceeding the configured NM vector length are not valid.

The Register Contents are updated at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

These bits are cleared when leaving CONFIG state or when entering STARTUP state.

18.2.8 Message Buffer Control Registers

18.2.8.1 FLXAnFRMRC — FlexRay Message RAM Configuration Register

The Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO.

The Message RAM can be divided into up three different areas; Static Buffer area, Static and Dynamic Buffer area, FIFO area. If present, the Static Buffer area is starting at Message Buffer 0.

The start of the Static and Dynamic Buffer area is configured by bits FLXAnFRMRC.FDB[7:0]. Bits FLXAnFRMRC.FDB[7:0] define the end of the Static Buffer area. If no Static Buffer area is present, the Static and Dynamic Buffer area starts at Message Buffer 0.

The start of the FIFO area is configured by bits FLXAnFRMRC.FFB[7:0]. Bits FLXAnFRMRC.FFB[7:0] define the end of the previous area, which can be either the Static Buffer area or the Static and Dynamic Buffer area. If no Static Buffer area and no Static and Dynamic Buffer area is present, the FIFO area starts at Message Buffer 0.

With bits FLXAnFRMRC.LCB[7:0], the end of the last configured area is configured which can be the Static Buffer area, the Static and Dynamic Buffer area or the FIFO area.

Figure 18.2 shows an example configuration of the Message RAM where all there area are configured.

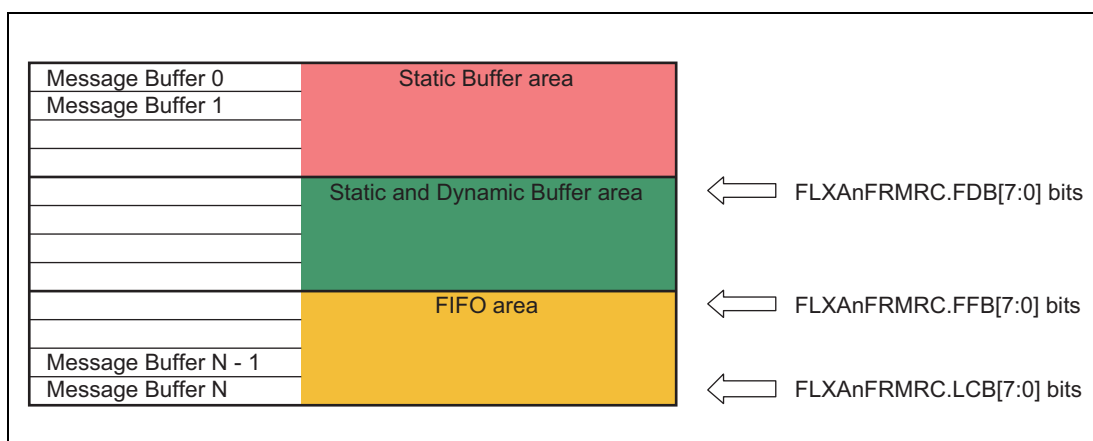


Figure 18.2 Message RAM Organization

CAUTIONS

1. In case the node is configured as sync node (bit FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (bit FLXAnFRSUCC1.TSM = 1), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 resp. 1 is treated like all other message buffers.
2. The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer. For details see Section 18.3.13, Message RAM.
3. In case two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the “Static Buffers” or at the beginning of the “Static + Dynamic Buffers” section.

4. The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non protocol conform configuration without a transmission slot in the static segment would still be operational.
5. The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via the FLXAnFRWRHS2.PLC[6:0] and FLXAnFRWRHS3.DP[6:0] bits. When the CC is not in DEFAULT_CONFIG or CONFIG state reconfiguration of message buffers belonging to the FIFO is locked.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0300_H

Value after reset: 0180 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----------|-----|-----|-----|-----|------|----------|----------|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | SPLM | SEC[1:0] | | LCB[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | FFB[7:0] | | | | | | | FDB[7:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.57 FLXAnFRMRC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 | SPLM | Sync Frame Payload Multiplex Bit 0: Only message buffer 0 locked against reconfiguration 1: Both message buffers 0 and 1 are locked against reconfiguration |
| 25, 24 | SEC[1:0] | Secure Buffers Bit 00: all buffers unlocked 01: static buffers locked, FIFO locked, limited transmission 10: all buffers locked 11: all buffers locked, limited transmission |
| 23 to 16 | LCB[7:0] | Last Configured Buffer Bit 0 to 127: Number of message buffers is FLXAnFRMRF.LCB + 1 128: No message buffer assigned to the FIFO |
| 15 to 8 | FFB[7:0] | First Buffer of FIFO Bit 0: All message buffers assigned to the FIFO 1 to 127: Message buffers from FLXAnFRMRC.FFB to FLXAnFRMRC.LCB assigned to the FIFO 128: No message buffer configured |
| 7 to 0 | FDB[7:0] | First Dynamic Buffer Bit 0: No group of message buffers exclusively for the static segment configured 1 to 127: Message buffers 0 to FLXAnFRMRC.FDB - 1 reserved for static segment 128: No dynamic message buffers configured |

(1) FLXAnFRMRC.SPLM

Sync Frame Payload Multiplex Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

This bit is only evaluated if the node is configured as sync node (bit FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (bit FLXAnFRSUCC1.TSM = 1).

When this bit is set to 1 message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B.

When this bit is set to 0, sync frames are transmitted from message buffer 0 with the same payload data on all channels configured. Note that the channel filter configuration for message buffer 0 resp. message buffer 1 has to be chosen according to this bit setting.

(2) FLXAnFRMRC.SEC

Secure Buffers Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state.

For temporary unlocking, see **Section 18.3.13.4, Host Handling of Access Errors**.

00_B = all buffers unlocked

Reconfiguration of message buffers enabled with numbers < FLXAnFRMRC.FFB enabled

Exception: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if bit FLXAnFRMRC.SPLM = 1, also message buffer 1) is always locked

01_B = static buffers locked, FIFO locked, limited transmission

Reconfiguration of message buffers with numbers < FLXAnFRMRC.FDB and with numbers ≥ FLXAnFRMRC.FFB locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

10_B = all buffers locked

Reconfiguration of all message buffers locked

11_B = all buffers locked, limited transmission

Reconfiguration of all message buffers locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

(3) FLXAnFRMRC.LCB

Last Configured Buffer Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (bits FLXAnFRMRC.FDB[7:0] < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FDB.

When a FIFO area is configured (bits FLXAnFRMRC.FFB[7:0] < 128), the user should configure bits FLXAnFRMRC.LCB[7:0] ≥ bits FLXAnFRMRC.FFB[7:0].

(4) FLXAnFRMRC.FFB

First Buffer of FIFO Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (bits FLXAnFRMRC.FDB[7:0] < 128), the user should configure bits FLXAnFRMRC.FFB[7:0] > bits FLXAnFRMRC.FDB[7:0].

(5) FLXAnFRMRC.FDB

First Dynamic Buffer Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

18.2.8.2 FLXAnFRFRF — FlexRay FIFO Rejection Filter Register

The FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0304_H

Value after reset: 0180 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|-----------|-----|-----|-----|-----|-----|----------|-----|-----|-----|-----|---------|-----|
| | — | — | — | — | — | — | — | RNF | RSS | CYF[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | FID[10:0] | | | | | | | | | | | CH[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.58 FLXAnFRFRF Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 25 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 24 | RNF | Reject Null Frames Bit 0: Null frames are stored in the FIFO 1: Reject all null frames |
| 23 | RSS | Reject in Static Segment Bit 0: FIFO also used for static segment 1: Reject messages in static segment |
| 22 to 16 | CYF[6:0] | Cycle Counter Filter Bit |
| 15 to 13 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 12 to 2 | FID[10:0] | Frame ID Filter Bit 0 to 2047: Frame ID filter values |
| 1, 0 | CH[1:0] | Channel Filter Bit 00: receive on both channels 01: receive only on channel B 10: receive only on channel A 11: no reception |

(1) FLXAnFRFRF.RNF

Reject Null Frames Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

If this bit is set to 1, received null frames are not stored in the FIFO.

(2) FLXAnFRFRF.RSS

Reject in Static Segment Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

If this bit is set to 1, the FIFO is used only for the dynamic segment.

(3) FLXAnFRFRF.CYF

Cycle Counter Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected. For details about the configuration of the cycle counter filter see **Section 18.3.8.2, Cycle Counter Filtering**.

(4) FLXAnFRFRF.FID

Frame ID Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FLXAnFRFRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When bits FLXAnFRFRFM.MFID[10:0] are zero, a frame ID filter value of zero means that no frame ID is rejected.

(5) FLXAnFRFRF.CH

Channel Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

18.2.8.3 FLXAnFRFRFM — FlexRay FIFO Rejection Filter Mask Register

The FlexRay FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering. If a bit is set to 1, it indicates that the corresponding bit in the FLXAnFRFRF register will not be considered for rejection filtering.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0308_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| | — | — | — | MFID[10:0] | | | | | | | | | | — | — | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 18.59 FLXAnFRFRFM Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 13 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 12 to 2 | MFID[10:0] | Mask Frame ID Filter Bit 0: Corresponding frame ID filter bit is used for rejection filtering 1: Ignore corresponding frame ID filter bit. |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(1) FLXAnFRFRFM.MFID

Mask Frame ID Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

18.2.8.4 FLXAnFRFCL — FlexRay FIFO Critical Level Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 030C_H

Value after reset: 0000 0080_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|---------|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | CL[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.60 FLXAnFRFCL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 to 0 | CL[7:0] | Critical Level Bit Critical Level |

(1) FLXAnFRFCL.CL

Critical Level Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

When the receive FIFO fill level (bits FLXAnFRCCSV.POCS[5:0]) FLXAnFRFSR.RFFL is equal or greater than the critical level configured by bits FLXAnFRFCL.CL[7:0], the receive FIFO critical level flag FLXAnFRFSR.RFCL is set to 1.

If FLXAnFRFCL.CL is programmed to values > 128, bit FLXAnFRFSR.RFCL is never set to 1.

18.2.9 Message Buffer Status Registers

18.2.9.1 FLXAnFRMHDS — FlexRay Message Handler Status Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0310_H

Value after reset: 0000 0080_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----------|----|----|----|----|----|------|----------|------|-------|-------|-----|-----|----|----|
| | — | MBU[6:0] | | | | | | — | MBT[6:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | FMB[6:0] | | | | | | CRAM | MFMB | FMBD | ATBF2 | ATBF1 | AMR | — | — | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R | R |

Table 18.61 FLXAnFRMHDS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 30 to 24 | MBU[6:0] | Message Buffer Updated Flag |
| 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | MBT[6:0] | Message Buffer Transmitted Flag |
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 14 to 8 | FMB[6:0] | Faulty Message Buffer Number Flag |
| 7 | CRAM | Clear all internal RAM's Flag 0: No execution of the CHI command CLEAR_RAMs 1: Execution of the CHI command CLEAR_RAMs ongoing |
| 6 | MFMB | Multiple Faulty Message Buffer Detection Flag 0 = No additional faulty message buffer. 1 = Additional faulty message buffer was detected while the FMBD flag is set to 1. |
| 5 | FMBD | Faulty Message Buffer Detection Flag 0 = No faulty message buffer. 1 = Message buffer referenced by bit FLXAnFRMHDS.FMB holds faulty data with a parity error. |
| 4 | ATBF2 | TBFRAM B Access Error Flag 0 = No access error 1 = Access error occurred when reading the RAM B. |
| 3 | ATBF1 | TBFRAM A Access Error Flag 0 = No access error. 1 = Access error occurred when reading the RAM A. |
| 2 | AMR | Message RAM Access Error Flag 0 = No access error 1 = Access error occurred when reading the Message RAM. |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(1) FLXAnFRMHDS.MBU

Message Buffer Updated Flag

Number of message buffer that was updated last by the CC. For this message buffer the respective ND and / or MBC flag in the FLXAnFRNDAT1 to FLXAnFRNDAT4 registers and the FLXAnFRMBSC1 to FLXAnFRMBSC4 registers are also set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(2) FLXAnFRMHDS.MBT

Message Buffer Transmitted Flag

Number of last successfully transmitted message buffer.

If the message buffer is configured for single-shot mode, the respective TXR flag in the FLXAnFRTXRQ1 to FLXAnFRTXRQ4 registers was reset to 0.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(3) FLXAnFRMHDS.FMB

Faulty Message Buffer Number Flag

This flag indicates that an access error occurred when reading from the message buffer referenced by FLXAnFRMHDS.FMB.

The value of this flag is only valid when one of flags FLXAnFRMHDS.AMR, FLXAnFRMHDS.ATBF1, FLXAnFRMHDS.ATBF2, and FLXAnFRMHDS.FMBD is set to 1.

This flag is not updated while the FLXAnFRMHDS.FMBD flag is 1.

This flag is cleared by the CHI command CLEAR_RAMs.

(4) FLXAnFRMHDS.CRAMP

Internal RAM Clear Flag

This flag indicates that the CHI command CLEAR_RAMs is ongoing (all bits of the message RAM, input buffer, output buffer and TBF are written to 0).

This flag is set by the CHI command CLEAR_RAMs.

(5) FLXAnFRMHDS.MFMB

Multiple Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This bit indicates that an additional faulty message buffer was detected while the FLXAnFRMHDS.FMBD flag is set.

This bit is cleared by the CHI command CLEAR_RAMs.

(6) FLXAnFRMHDS.FMBD

Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This bit indicates that the message buffer referenced by FLXAnFRMHDS.FMB holds faulty data due to an access error.

This bit is cleared by the CHI command CLEAR_RAMs.

(7) FLXAnFRMHD.ATBF2

TBFRAM B Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the TBFRAM B.

CAUTION

When this flag changes from 0 to 1, the FLXAnFREIR.AERR bit is set to 1. This flag can be reset by the CHI command CLEAR_RAMs.

(8) FLXAnFRMHD.ATBF1

TBFRAM A Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the TBFRAM A.

CAUTION

When this flag changes from 0 to 1, the FLXAnFREIR.AERR bit is set to 1. This flag can be reset by the CHI command CLEAR_RAMs.

(9) FLXAnFRMHDS.AMR

Message RAM Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the Message RAM.

CAUTION

When this flag changes from 0 to 1, the FLXAnFREIR.AERR bit is set to 1. This flag can be reset by the CHI command CLEAR_RAMs.

18.2.9.2 FLXAnFRLDTS — FlexRay Last Dynamic Transmit Slot Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0314_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | LDTB[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | LDTA[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.62 FLXAnFRLDTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 27 | Reserved | When read, the value after reset is read. |
| 26 to 16 | LDTB[10:0] | Last Dynamic Transmission Channel B Flag |
| 15 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 0 | LDTA[10:0] | Last Dynamic Transmission Channel A Flag |

(1) FLXAnFRLDTS.LDTB

Last Dynamic Transmission Channel B Flag

Stores the value of vSlotCounter[B] at the time of the last frame transmission on channel B in the dynamic segment of this node.

It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs of channel B.

(2) FLXAnFRLDTS.LDTA

Last Dynamic Transmission Channel A Flag

Stores the value of vSlotCounter[A] at the time of the last frame transmission on channel A in the dynamic segment of this node.

It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs of channel A.

18.2.9.3 FLXAnFRFSR — FlexRay FIFO Status Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0318_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|----|-----|------|------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RFFL[7:0] | | | | | | | — | — | — | — | — | RFO | RFCL | RFNE | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.63 FLXAnFRFSR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. |
| 15 to 8 | RFFL[7:0] | Receive FIFO Fill Level Flag |
| 7 to 3 | Reserved | When read, the value after reset is read. |
| 2 | RFO | Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected |
| 1 | RFCL | Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached |
| 0 | RFNE | Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty |

(1) FLXAnFRFSR.RFFL

Receive FIFO Fill Level Flag

Indicates the number of FIFO buffers filled with new data not yet read by the Host. Maximum value is 128.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(2) FLXAnFRFSR.RFO

Receive FIFO Overrun Flag

The flag is set to 1 by the CC when a receive FIFO overrun is detected.

When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, the FLXAnFREIR.RFO flag is set to 1.

The flag is cleared by the next FIFO read access issued by the Host.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(3) FLXAnFRFSR.RFCL

Receive FIFO Critical Level Flag

This flag is set to 1 when the value of the receive FIFO fill level FLXAnFRFSR.RFFL[7:0] is equal or greater than the critical level as configured by FLXAnFRFCL.CL.

When this bit changes from 0 to 1 bit FLXAnFRSIR.RFCL is set to 1, and if enabled, an interrupt is generated.

The flag is cleared by the CC as soon as the value of the FLXAnFRFSR.RFFL[7:0] bits drops below the value set in the FLXAnFRFCL.CL bit.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(4) FLXAnFRFSR.RFNE

Receive FIFO Not Empty Flag

This flag is set to 1 by the CC when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, the FLXAnFRSIR.RFNE bit is set to 1.

The bit is reset to 0 after the Host has read all messages from the FIFO.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

18.2.9.4 FLXAnFRMHDF — FlexRay Message Handler Constraints Flags Register

Some constraints exist for the Message Handler regarding bus clock frequency, Message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in the FLXAnFRMHDF.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 031C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | WAHP | TNSB | TNSA | TBFB | TBFA | FNFB | FNFA | SNUB | SNUA |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.64 FLXAnFRMHDF Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | WAHP | Write Attempt to Header Partition Flag 0: No write attempt to header partition 1: Write attempt to header partition |
| 7 | TNSB | Transmission Not Started Channel B Flag 0: No transmission not started on channel B 1: Transmission not started on channel B |
| 6 | TNSA | Transmission Not Started Channel A Flag 0: No transmission not started on channel A 1: Transmission not started on channel A |
| 5 | TBFB | TBFB Access Failure B Flag 0: No TBFB access failure 1: TBFB access failure |
| 4 | TBFA | Temporary buffer Access Failure A Flag 0: No TBFA access failure 1: TBFA access failure |
| 3 | FNFB | Find Sequence Not Finished Channel B Flag 0: No find sequence not finished for channel B 1: Find sequence not finished for channel B |
| 2 | FNFA | Find Sequence Not Finished Channel A Flag 0: No find sequence not finished for channel A 1: Find sequence not finished for channel A |
| 1 | SNUB | Status Not Updated Channel B Flag 0: No overload condition occurred when updating MBS (FLXAnFRMBS) for channel B 1: Message buffer status (FLXAnFRMBS) for channel B not updated |
| 0 | SNUA | Status Not Updated Channel A Flag 0: No overload condition occurred when updating MBS (FLXAnFRMBS) for channel A 1: Message buffer status (FLXAnFRMBS) for channel A not updated |

(1) FLXAnFRMHDF.WAHP

Write Attempt to Header Partition Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Outside DEFAULT_CONFIG and CONFIG state this flag is set to 1 by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses.

When this flag changes from 0 to 1, the FLXAnFREIR.MHF flag is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(2) FLXAnFRMHDF.TNSB

Transmission Not Started Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler was not ready to start a scheduled transmission on channel B at the action point of the configured slot.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(3) FLXAnFRMHDF.TNSA

Transmission Not Started Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Message Handler was not ready to start a scheduled transmission on channel A at the action point of the configured slot.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(4) FLXAnFRMHDF.TBFB

TBFB Access Failure Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when a read or write access to TBFB requested by PRT (Protocol controller) B could not complete within the available time.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(5) FLXAnFRMHDF.TBFA

TBFA Access Failure Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when a read or write access to TBFA requested by PRT A could not complete within the available time.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(6) FLXAnFRMHDF.FNFB

Find sequence Not Finished flag, channel B

Writing 0 has no effect on the bit's value.

Writing a 1 to the bit clears it.

This flag is set to 1 by the CC when the message handler, due to being overloaded, was not able to finish a find sequence (scan of message RAM for a matching message buffer).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is also set to 1.

The value is reset on leaving the CONFIG state or entering the STARTUP state.

The value is also reset by the CLEAR_RAMs CHI command.

(7) FLXAnFRMHDF.FNFA

Find sequence Not Finished flag, channel A

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(8) FLXAnFRMHDF.SNUB

Status Not Updated Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status (FLXAnFRMBS).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

(9) FLXAnFRMHDF.SNUA

Status Not Updated Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status (FLXAnFRMBS).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMs.

18.2.9.5 FLXAnFRTXRQm — FlexRay Transmission Request m (m = 1 to 4)

The four registers reflect the state of the TXR flags of all configured message buffers. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 128, the remaining TXR flags have no meaning.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0320_H to <FLXAn_base> + 032C_H (<FLXAn_base> + 0320_H + (m-1) × 4)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| TXRp[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXRp[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.65 FLXAnFRTXRQm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-----------------------------|
| 31 to 0 | TXRp[31:0] | Transmission Request Flag p |

(1) FLXAnFRTXRQm.TXRp (p = (m - 1) × 32 to (m × 32 - 1))

Transmission Request Flag p

If the flag is set to 1, the respective message buffer is ready for transmission respectively transmission of this message buffer is in progress.

In single-shot mode the flags are reset to 0 after transmission has completed.

This bit is cleared by the CHI command CLEAR_RAMs.

18.2.9.6 FLXAnFRNDATm — FlexRay New Data Register m (m = 1 to 4)

The four registers reflect the state of the ND flags of all configured message buffers. ND flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, the remaining ND flags have no meaning.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0330_H to <FLXAn_base> + 033C_H (<FLXAn_base> + 0330_H + (m-1) × 4)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| NDp[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NDp[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.66 FLXAnFRNDATm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|-----------------|
| 31 to 0 | NDp[31:0] | New Data Flag p |

(1) FLXAnFRNDATm.NDp (p = (m - 1) × 32 to (m × 32 - 1))

New Data Flag p

The flags are set to 1 when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer.

The flags are not set to 1 after reception of null frames except for message buffers belonging to the receive FIFO.

An ND flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

Reset when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMs.

18.2.9.7 FLXAnFRMBSCm — FlexRay Message Buffer Status Changed Register m (m = 1 to 4)

The four registers reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 128, the remaining MBC flags have no meaning.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0340_H to <FLXAn_base> + 034C_H (<FLXAn_base> + 0340_H + (m-1) × 4)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| MBCp[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MBCp[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.67 FLXAnFRMBSCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--------------------------------------|
| 31 to 0 | MBCp[31:0] | Message Buffer Status Changed Flag p |

(1) FLXAnFRMBSCm.MBCp (p = (m - 1) × 32 to (m × 32 - 1))

Message Buffer Status Changed Flag p

Indicates whether the Message Handler has changed one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, FTB in the header section (see **Section 18.2.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** and **Section 18.3.13.1, Header Partition**) of the respective message buffer.

An MBC flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

Reset when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMs.

18.2.10 Input Buffer

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the Message Buffer Status as described in **Section 18.2.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers only the payload length configured and the data pointer need to be configured via bits FLXAnFRWRHS2.PLC[6:0] and bits FLXAnFRWRHS3.DP[10:0]. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in detail in **Section 18.3.12.2, Host access to Message RAM**.

These registers cannot be written when the input data transfer function shown in **Section 18.3.16.1, Input Data Transfer** is used and the FLXAnFRITS.ITS bit is 1.

18.2.10.1 FLXAnFRWRDSm — FlexRay Write Data Section Register m (m = 1 to 64)

This register holds the data words to be transferred to the data section of the specified message buffer. The number of data words written to the Message RAM is defined by the payload length configured in bits FLXAnFRWRHS2.PLC[6:0].

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0400_H to <FLXAn_base> + 04FC_H(<FLXAn_base> + 0400_H + (m-1) × 4)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | MD[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MD[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.68 FLXAnFRWRDSm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|------------------|
| 31 to 0 | MD[31:0] | Message Data Bit |

(1) FLXAnFRWRDSm.MD

Message Data Bit

Reset by the CHI command CLEAR_RAMs.

CAUTIONS

1. In case the FLXAnFRWRHS2.PLC[6:0] bits specifies an odd payload length, the remaining message data bytes are unused.
2. When writing to the FLXAnFRWRDSm register, all bytes have to be written by one 32-bit access, two consecutive 16-bit accesses, or four consecutive 8-bit accesses before the transfer from the Input Buffer to the Message RAM is started. If not all bytes of a 32-bit word have been written by the Host (8/16-bit access only), the FLXAnFRWRDSm register holds partly undefined data.

18.2.10.2 FLXAnFRWRHS1 — FlexRay Write Header Section Register 1

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0500_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|-----|-----|------|-----------|---------|-----|-----|----------|-----|-----|-----|-----|-----|-----|
| | — | — | MBI | TXM | PPIT | CFG | CH[1:0] | | — | CYC[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | FID[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.69 FLXAnFRWRHS1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 | MBI | Message Buffer Interrupt Bit 0: The corresponding message buffer interrupt is disabled 1: The corresponding message buffer interrupt is enabled |
| 28 | TXM | Transmission Mode Setting Bit 0: Continuous mode 1: Single-shot mode |
| 27 | PPIT | Payload Preamble Indicator Transmit Bit 0: Payload Preamble Indicator is set to 0 1: Payload Preamble Indicator is set to 1 |
| 26 | CFG | Message Buffer Direction Configuration Bit 0: The corresponding buffer is configured as Receive Buffer 1: The corresponding buffer is configured as Transmit Buffer |
| 25,24 | CH[1:0] | Channel Filter Control Bit |
| 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | CYC[6:0] | Cycle Code Bit |
| 15 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 to 0 | FID[10:0] | Frame ID Bit |

(1) FLXAnFRWRHS1.MBI

Message Buffer Interrupt Enable Bit

This bit enables the message buffer interrupt.

After a dedicated receive buffer has been updated by the Message Handler, bit FLXAnFRSIR.RXI and /or bit FLXAnFRSIR.MBSI are set to 1. After a transmission has completed bit FLXAnFRSIR.TXI is set to 1.

(2) FLXAnFRWRHS1.TXM

Transmission Mode Setting Bit

This bit selects transmit mode of the corresponding message buffer. For transmit mode, see **Section 18.3.9.3, Transmit Buffers**.

(3) FLXAnFRWRHS1.PPIT

Payload Preamble Indicator Transmit Bit

This bit is used to control the state of the Payload Preamble Indicator in transmit frames in transmit frames of the corresponding message buffer.

If the bit is set to 1 in a static message buffer, the respective message buffer holds network management information.

If the bit is set to 1 in a dynamic message buffer the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the FlexRay module, but can be done by the Host.

(4) FLXAnFRWRHS1.CFG

Message Buffer Direction Configuration Bit

This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO the bit is not evaluated.

When an unused 32-bit or larger area is not allocated to the head of the data partition, set the data section of the message buffer that is allocated immediately after the header partition (of the last buffer) as a transmit buffer by setting this bit to 1.

(5) FLXAnFRWRHS1.CH

Channel Filter Control Bit

The 2-bit channel filtering field associated with each buffer serves as a filter for receive buffers, and as a control field for transmit buffers.

| CH[1:0] | Transmit Buffer transmit frame on | Receive Buffer store frame received from |
|---------|-------------------------------------|--|
| 00 | No transmission | Ignore frame |
| 01 | Channel A | Channel A |
| 10 | Channel B | Channel B |
| 11 | Both channels (static segment only) | Channel A or B (store first semantically valid frame; static segment only) |

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted resp. received frames are ignored (same function as CH = "00_B")

(6) FLXAnFRWRHS1.CYC

Cycle Code Bit

The 7-bit cycle code determines the cycle set used for cycle counter filtering.

For details about the configuration of the cycle code **Section 18.3.8.2, Cycle Counter Filtering**.

(7) FLXAnFRWRHS1.FID

Frame ID Bit

Frame ID of the selected message buffer. The frame ID defines the slot number for transmission / reception of the respective message.

Message buffers with frame ID = 0 are considered as not valid.

18.2.10.3 FLXAnFRWRHS2 — FlexRay Write Header Section Register 2

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0504_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|-----------|-----|-----|-----|----------|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | PLC[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | CRC[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.70 FLXAnFRWRHS2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | PLC[6:0] | Payload Length Configured Bit |
| 15 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 to 0 | CRC[10:0] | Header CRC Bit (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required Transmit Buffer: Header CRC is configured |

(1) FLXAnFRWRHS2.PLC

Payload Length Configured Bit

Length of data section (number of 2-byte words) as configured by the Host.

During static segment the static frame payload length as configured by bits

FLXAnFRMHDC.SFDL[6:0] defines the payload length for all static frames. If the payload length configured by bits FLXAnFRWRHS2.PLC[6:0] is shorter than this value padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is “0000_H” (see **Section 18.3.9.3, Transmit Buffers**).

(2) FLXAnFRWRHS2.CRC

Header CRC Bit (vRF!Header!HeaderCRC)

Setting of the receive buffer is not required.

Transmitting of the message buffer needs the header CRC calculation and setting.

For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by bits FLXAnFRMHDC.SFDL[6:0].

18.2.10.4 FLXAnFRWRHS3 — FlexRay Write Header Section Register 3

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0508_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | DP[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.71 FLXAnFRWRHS3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 to 0 | DP[10:0] | Data Pointer Bit |

(1) FLXAnFRWRHS3.DP

Data Pointer Bit

Configures the pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

18.2.10.5 FLXAnFRIBCM — FlexRay Input Buffer Command Mask Register

Configures how the message buffer in the Message RAM selected by register FLXAnFRIBCR is updated. When IBF Host and IBF Shadow are swapped, also mask bits FLXAnFRIBCM.LHSH, FLXAnFRIBCM.LDSH, and FLXAnFRIBCM.STXRH are swapped with bits FLXAnFRIBCM.LHSS, FLXAnFRIBCM.LDSS, and FLXAnFRIBCM.STXRS.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0510_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|------|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | STXRS | LDSS | LHSS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | STXRH | LDSH | LHSH |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 18.72 FLXAnFRIBCM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | STXRS | Set Transmission Request Shadow Flag 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission (operation ongoing or finished) |
| 17 | LDSS | Load Data Section Shadow Flag 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished) |
| 16 | LHSS | Load Header Section Shadow Flag 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished) |
| 15 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | STXRH | Set Transmission Request Host Bit 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission |
| 1 | LDSH | Load Data Section Host Bit 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM |
| 0 | LHSH | Load Header Section Host Bit 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM |

(1) FLXAnFRIBCM.STXRS

Set Transmission Request Shadow Flag

(2) FLXAnFRIBCM.LDSS

Load Data Section Shadow Flag

(3) FLXAnFRIBCM.LHSS

Load Header Section Shadow Flag

(4) FLXAnFRIBCM.STXRH

Set Transmission Request Host Bit

If this bit is set to 1, the TXR flag for the selected message buffer is set in the FLXAnFRTXRQ1 to FLXAnFRTXRQ4 registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the CC after transmission has completed.

TXR is evaluated for transmit buffers only.

(5) FLXAnFRIBCM.LDSH

Set Load Data Section Host Bit

(6) FLXAnFRIBCM.LHSH

Set Load Header Section Host Bit

18.2.10.6 FLXAnFRIBCR — FlexRay Input Buffer Command Request Register

When the Host writes the number of the target message buffer in the Message RAM to bits FLXAnFRIBCR.IBRH[6:0], IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under bits FLXAnFRIBCR.IBRH[6:0] and bits FLXAnFRIBCR.IBRS[6:0] are also swapped (see **Section 18.3.12.2 (1), Data Transfer from Input Buffer to Message RAM**).

With this write operation bit FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by bits FLXAnFRIBCR.IBRS[6:0].

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message into the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to bits FLXAnFRIBCR.IBRH[6:0].

If a write access to bits FLXAnFRIBCR.IBRH[6:0] occurs while bit FLXAnFRIBCR.IBSYS is 1, bit FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, bit FLXAnFRIBCR.IBSYH is reset to 0. Bit FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under bits FLXAnFRIBCR.IBRH[6:0] and bits FLXAnFRIBCR.IBRS[6:0] are also swapped.

Any write access to an Input Buffer register while both bit FLXAnFRIBCR.IBSYS and bit FLXAnFRIBCR.IBSYH are set to 1 will cause the error flag FLXAnFREIR.IIBA to be set to 1.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0514_H

Value after reset: 00000000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|----|----|----|----|----|----|----|----|-----------|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | IBSYS | — | — | — | — | — | — | — | — | IBRS[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IBSYH | — | — | — | — | — | — | — | — | IBRH[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.73 FLXAnFRIBCR Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 | IBSYS | Input Buffer Busy Shadow Flag 0: Transfer between IBF Shadow and Message RAM completed 1: Transfer between IBF Shadow and Message RAM in progress |
| 30 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | IBRS[6:0] | Input Buffer Request Shadow Flag |
| 15 | IBSYH | Input Buffer Busy Host Flag 0: No request pending 1: Request while transfer between IBF Shadow and Message RAM in progress |

Table 18.73 FLXAnFRIBCR Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 14 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | IBRH[6:0] | Input Buffer Request Host Bit |

(1) FLXAnFRIBCR.IBSYS

Input Buffer Busy Shadow Flag

Set to 1 after writing bits FLXAnFRIBCR.IBRH[6:0].

This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.

When the transfer between IBF Shadow and the Message RAM has completed, FLXAnFRIBCR.IBSYS is set back to 0.

(2) FLXAnFRIBCR.IBRS

Input Buffer Request Shadow Flag

Number of the target message buffer actually updated / lately updated.

(3) FLXAnFRIBCR.IBSYH

Input Buffer Busy Host Flag

Set to 1 by writing bits FLXAnFRIBCR.IBRH[6:0] while bit FLXAnFRIBCR.IBSYS is still 1.

This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.

After the ongoing transfer between IBF Shadow and the Message RAM has completed, the FLXAnFRIBCR.IBSYH is set back to 0.

(4) FLXAnFRIBCR.IBRH

Input Buffer Request Host Bit

Selects the target message buffer in the Message RAM for data transfer from Input Buffer.

18.2.11 Output Buffer

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in **Section 18.3.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

These registers cannot be written when the output data transfer function shown in **Section 18.3.16.2, Output Data Transfer**, in Output Data Transfer is used and the FLXAnFROTS.OTS bit is 1.

18.2.11.1 FLXAnFRRDDSm — FlexRay Read Data Section Register m (m = 1 to 64)

Holds the data words read from the data section of the addressed message buffer. This register holds the data words to be transferred to the data section of the specified message buffer. The number of data words (DWN) read from the Message RAM is defined by the payload length configured in bits FLXAnFRRDHS2.PLC[6:0].

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0600_H to <FLXAn_base> + 06FC_H (<FLXAn_base> + 0600_H + (m-1) × 4)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | MD[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MD[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.74 FLXAnFRRDDSm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--------------|
| 31 to 0 | MD[31:0] | Message Data |

(1) FLXAnFRRDDSm.MD

Message Data Flag

Cleared by the CHI command CLEAR_RAMs.

CAUTION

In case the FLXAnFRWRHS2.PLC[6:0] bits specifies an odd payload length, the remaining message data bytes are unused.

18.2.11.2 FLXAnFRRDHS1 — FlexRay Read Header Section Register 1

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0700_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|-----|-----|------|-----------|---------|----|----|----------|----|----|----|----|----|----|
| | — | — | MBI | TXM | PPIT | CFG | CH[1:0] | | — | CYC[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | FID[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.75 FLXAnFRRDHS1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31, 30 | Reserved | When read, the value after reset is read. |
| 29 | MBI | Message Buffer Interrupt Flag |
| 28 | TXM | Transmission Mode Flag |
| 27 | PPIT | Payload Preamble Indicator Transmit Flag |
| 26 | CFG | Message Buffer Direction Configuration Flag |
| 25, 24 | CH[1:0] | Channel Filter Control Flag |
| 23 | Reserved | When read, the value after reset is read. |
| 22 to 16 | CYC[6:0] | Cycle Code |
| 15 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 0 | FID[10:0] | Frame ID |

(1) FLXAnFRRDHS1.MBI

Message Buffer Interrupt Flag

Values as configured by the Host via bit FLXAnFRWRHS1.MBI.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(2) FLXAnFRRDHS1.TXM

Transmission Mode Flag

Values as configured by the Host via bit FLXAnFRWRHS1.TXM.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(3) FLXAnFRRDHS1.PPIT

Payload Preamble Indicator Transmit Flag

Values as configured by the Host via bit FLXAnFRWRHS1.PPIT.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(4) FLXAnFRRDHS1.CFG

Message Buffer Direction Configuration Flag

Values as configured by the Host via bit FLXAnFRWRHS1.CFG.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(5) FLXAnFRRDHS1.CH

Channel Filter Control Flag

Values as configured by the Host via bit FLXAnFRWRHS1.CH.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(6) FLXAnFRRDHS1.CYC

Cycle Code

Values as configured by the Host via bit FLXAnFRWRHS1.CYC.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(7) FLXAnFRRDHS1.FID

Frame ID

Values as configured by the Host via bit FLXAnFRWRHS1.FID.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are holding the received frame ID.

18.2.11.3 FLXAnFRRDHS2 — FlexRay Read Header Section Register 2

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0704_H

Value after reset: 0000 0000_H

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, the FLXAnFRWRHS2 register is updated from data frames only.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----------|----|----|----|-----------|----|----|----------|----|----|----|----|----|----|----|
| | — | PLR[6:0] | | | | | | — | PLC[6:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | CRC[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.76 FLXAnFRRDHS2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 | Reserved | When read, the value after reset is read. |
| 30 to 24 | PLR[6:0] | Payload Length Received Flag (vRF!Header!Length) |
| 23 | Reserved | When read, the value after reset is read. |
| 22 to 16 | PLC[6:0] | Payload Length Configuration Flag |
| 15 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 0 | CRC[10:0] | Header CRC Flag (vRF!Header!HeaderCRC) |

(1) FLXAnFRRDHS2.PLR

Payload Length Received Flag (vRF!Header!Length)

Payload length (vRF!Header!Length) value updated from received data frames (exception: if message buffer belongs to the receive FIFO FLXAnFRRDHS2.PLR is also updated from received null frames).

(2) FLXAnFRRDHS2.PLC

Payload Length Configuration Flag

Length of data section (number of 2-byte words) as configured by the Host.

(3) FLXAnFRRDHS2.CRC

Header CRC Flag (vRF!Header!HeaderCRC)

Receive Buffer: Header CRC (vRF!Header!HeaderCRC) updated from received data frames

Transmit Buffer: Header CRC configured by the Host

(4) Data storage

When a message is stored into a message buffer the following behavior with respect to payload length received and payload length configured is implemented:

Bits $FLXAnFRRDHS2.PLR[6:0] > Bits\ FLXAnFRRDHS2.PLC[6:0]$:

The payload data stored in the message buffer is truncated to the payload length configured if bits $FLXAnFRRDHS2.PLC[6:0]$ even or else truncated to bits $FLXAnFRRDHS2.PLC[6:0] + 1$.

Bits $FLXAnFRRDHS2.PLR[6:0] \leq Bits\ FLXAnFRRDHS2.PLC[6:0]$:

The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by bits $FLXAnFRRDHS2.PLC[6:0]$ are filled with undefined data.

Bits $FLXAnFRRDHS2.PLR[6:0] = zero$:

The message buffer's data section is filled with undefined data

Bits $FLXAnFRRDHS2.PLC[6:0] = zero$:

Message buffer has no data section configured. No data is stored into the message buffer's data section.

CAUTIONS

1. The Message RAM is organized in 4-byte words. When received data is stored into a message buffer's data section, the number of 2-byte data words written into the message buffer is the $FLXAnFRRDHS2.PLC[6:0]$ bits rounded to the next even value.
2. The $FLXAnFRRDHS2.PLC[6:0]$ bits should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only. For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, the $FLXAnFRWRHS2$ register is updated from data frames only.

18.2.11.4 FLXAnFRRDHS3 — FlexRay Read Header Section Register 3

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, the FLXAnFRWRHS3 register is updated from data frames only.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0708_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|-----|-----|-----|----------|-----|-----|----|----|----------|----|----|----|----|----|
| | — | — | RES | PPI | NFI | SYN | SFI | RCI | — | — | RCC[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | DP[10:0] | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.77 FLXAnFRRDHS3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31, 30 | Reserved | When read, the value after reset is read. |
| 29 | RES | Reserved Bit Indicator Flag (vRF!Header!Reserved) |
| 28 | PPI | Payload Preamble Indicator (vRF!Header!PPIIndicator) |
| 27 | NFI | Null Frame Indicator Flag (vRF!Header!NFIIndicator) 0: Up to now no data frame has been stored into the respective message buffer 1: At least one data frame has been stored into the respective message buffer |
| 26 | SYN | Sync Frame Indicator Flag (vRF!Header!SyFIndicator) 0: The received frame is not a sync frame 1: The received frame is a sync frame |
| 25 | SFI | Startup Frame Indicator Flag (vRF!Header!SuFIndicator) 0: The received frame is not a startup frame 1: The received frame is a startup frame |
| 24 | RCI | Received on Channel Indicator Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A |
| 23, 22 | Reserved | When read, the value after reset is read. |
| 21 to 16 | RCC[5:0] | Receive Cycle Counter (vRF!Header!CycleCount) |
| 15 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 0 | DP[10:0] | Data Pointer Flag |

(1) FLXAnFRRDHS3.RES

Reserved Bit Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.

(2) FLXAnFRRDHS3.PPI

Payload Preamble Indicator Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

0 = The payload segment of the received frame does not contain a network management vector nor a message ID

1 = Static segment: Network management vector in the first part of the payload
Dynamic segment: Message ID in the first part of the payload

(3) FLXAnFRRDHS3.NFI

Null Frame Indicator Flag (vRF!Header!NFIndicator)

Is set to 1 after storage of the first received data frame.

(4) FLXAnFRRDHS3.SYN

Sync Frame Indicator Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.

(5) FLXAnFRRDHS3.SFI

Startup Frame Indicator Flag (vRF!Header!SuFIndicator)

A startup frame is marked by the startup frame indicator.

(6) FLXAnFRRDHS3.RCI

Received on Channel Indicator Flag (vSS!Channel)

Indicates the channel from which the received data frame was taken to update the respective receive buffer.

(7) FLXAnFRRDHS3.RCC

Receive Cycle Counter (vRF!Header!CycleCount)

Cycle counter value updated from received data frame.

(8) FLXAnFRRDHS3.DP

Data Pointer Flag

Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

The bit value is the same as that set in bit FLXAnFRWRHS3.DP

18.2.11.5 FLXAnFRMBS — FlexRay Message Buffer Status Register

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer.

The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated.

The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all FLXAnFRMBS flags are reset to zero independent of which FLXAnFRIBCM bits are set or not.

For details about receive / transmit filtering see **Section 18.3.8, Filtering and Masking**, **Section 18.3.9, Transmit Process** and **Section 18.3.10, Receive Process**.

Whenever the Message Handler changes one of the flags FLXAnFRMBS.VFRA, FLXAnFRMBS.VFRB, FLXAnFRMBS.SEOA, FLXAnFRMBS.SEOB, FLXAnFRMBS.CEOA, FLXAnFRMBS.CEOB, FLXAnFRMBS.SVOA, FLXAnFRMBS.SVOB, FLXAnFRMBS.TCIA, FLXAnFRMBS.TCIB, FLXAnFRMBS.ESA, FLXAnFRMBS.ESB, FLXAnFRMBS.MLST, FLXAnFRMBS.FTA, FLXAnFRMBS.FTB the respective message buffer's MBC flag in registers FLXAnFRMBSC1 to FLXAnFRMBSC4 is set.

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 070C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----|-----|------|------|------|------|------|------|------|------|----------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | RESS | PPIS | NFIS | SYNS | SFIS | RCIS | — | — | CCS[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | FTB | FTA | — | MLST | ESB | ESA | TCIB | TCIA | SVOB | SVOA | CEOB | CEOA | SEOB | SEOA | VFRB | VFRA |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.78 FLXAnFRMBS Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31, 30 | Reserved | When read, the value after reset is read. |
| 29 | RESS | Reserved Bit Status Flag (vRF!Header!Reserved) |
| 28 | PPIS | Payload Preamble Indicator Status Flag (vRF!Header!PPIIndicator) 0: PPI indicator set to 0 1: PPI indicator set to 1 |
| 27 | NFIS | Null Frame Indicator Status Flag (vRF!Header!NFIndicator) 0: Received frame is a null frame 1: Received frame is not a null frame |
| 26 | SYNS | Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator) 0: No sync frame received 1: The received frame is a sync frame |

Table 18.78 FLXAnFRMBS Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 25 | SFIS | Startup Frame Indicator Status Flag (vRF!Header!SuIndicator) 0: No startup frame received 1: The received frame is a startup frame |
| 24 | RCIS | Received on Channel Indicator Status Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A |
| 23, 22 | Reserved | When read, the value after reset is read. |
| 21 to 16 | CCS[5:0] | Cycle Count Status Flag |
| 15 | FTB | Frame Transmitted on Channel B Flag 0: No data frame transmitted on channel B 1: Data frame transmitted on channel B |
| 14 | FTA | Frame Transmitted on Channel A Flag 0: No data frame transmitted on channel A 1: Data frame transmitted on channel A |
| 13 | Reserved | When read, the value after reset is read. |
| 12 | MLST | Message Lost Flag 0: No message lost 1: Unprocessed message was overwritten |
| 11 | ESB | Empty Slot Channel B Flag 0: Bus activity detected in the assigned slot on channel B 1: No bus activity detected in the assigned slot on channel B |
| 10 | ESA | Empty Slot Channel A Flag 0: Bus activity detected in the assigned slot on channel A 1: No bus activity detected in the assigned slot on channel A |
| 9 | TCIB | Transmission Conflict Indication Channel B Flag (vSS!TxConflictB) 0: No transmission conflict occurred on channel B 1: Transmission conflict occurred on channel B |
| 8 | TCIA | Transmission Conflict Indication Channel A Flag (vSS!TxConflictA) 0: No transmission conflict occurred on channel A 1: Transmission conflict occurred on channel A |
| 7 | SVOB | Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB) 0: No slot boundary violation observed on channel B 1: Slot boundary violation observed on channel B |
| 6 | SVOA | Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA) 0: No slot boundary violation observed on channel A 1: Slot boundary violation observed on channel A |
| 5 | CEOB | Content Error Observed on Channel B Flag (vSS!ContentErrorB) 0: No content error observed on channel B 1: Content error observed on channel B |
| 4 | CEOA | Content Error Observed on Channel A Flag (vSS!ContentErrorA) 0: No content error observed on channel A 1: Content error observed on channel A |
| 3 | SEOB | Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB) 0: No syntax error observed on channel B 1: Syntax error observed on channel B |
| 2 | SEOA | Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA) 0: No syntax error observed on channel A 1: Syntax error observed on channel A |
| 1 | VFRB | Valid Frame Received on Channel B (vSS!ValidFrameB) 0: No valid frame received on channel B 1: Valid frame received on channel B |
| 0 | VFRA | Valid Frame Received on Channel A Flag (vSS!ValidFrameA) 0: No valid frame received on channel A 1: Valid frame received on channel A |

(1) FLXAnFRMBS.RESS

Reserved Bit Status Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(2) FLXAnFRMBS.PPIS

Payload Preamble Indicator Status Flag (vRF!Header!PPIIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

0 = PPI indicator set to 0

The payload segment of the received frame does not contain a network management vector or a message ID

1 = PPI indicator set to 1

Static segment: Network management vector at the beginning of the payload

Dynamic segment: Message ID at the beginning of the payload

(3) FLXAnFRMBS.NFIS

Null Frame Indicator Status Flag (vRF!Header!NFIndicator)

If set to 0 the payload segment of the received frame contains no usable data.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(4) FLXAnFRMBS.SYNS

Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(5) FLXAnFRMBS.SFIS

Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator)

The startup frame indicator specifies a startup frame.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(6) FLXAnFRMBS.RCIS

Received on Channel Indicator Status Flag (vSS!Channel)

Indicates the channel on which the frame was received.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(7) FLXAnFRMBS.CCS

Cycle Count Status Flag

Actual cycle count when status was updated.

(8) FLXAnFRMBS.FTB

Frame Transmitted on Channel B Flag

Indicates that this node has transmitted a data frame in the configured slot on channel B.

CAUTION

The FlexRay protocol specification requires that this bit can only be reset by the Host. Therefore, the Cycle Count Status FLXAnFRMBS.CCS bit for this bit is only valid for the cycle where the bit is set to 1.

(9) FLXAnFRMBS.FTA

Frame Transmitted on Channel A Flag

Indicates that this node has transmitted a data frame in the configured slot on channel A.

CAUTION

The FlexRay protocol specification requires that this bit can only be reset by the Host. Therefore the Cycle Count Status FLXAnFRMBS.CCS bit for this bit is only valid for the cycle where this bit is set to 1.

(10) FLXAnFRMBS.MLST

Message Lost Flag

The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame.

Not affected by reception of null frames except for message buffers belonging to the receive FIFO. Bits FLXAnFRNDATm.NDp is reset to 0 by a Host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset to 0 by reading out the message buffer via OBF.

(11) FLXAnFRMBS.ESB

Empty Slot Channel B Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(12) FLXAnFRMBS.ESA

Empty Slot Channel A Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(13) FLXAnFRMBS.TCIB

Transmission Conflict Indication Channel B Flag (vSS!TxConflictB)

A transmission conflict indication is set to 1 if a transmission conflict has occurred on channel B.

(14) FLXAnFRMBS.TCIA

Transmission Conflict Indication Channel A Flag (vSS!TxConflictA)

A transmission conflict indication is set if a transmission conflict has occurred on channel A.

(15) FLXAnFRMBS.SVOB

Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel B.

(16) FLXAnFRMBS.SVOA

Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel A.

(17) FLXAnFRMBS.CEOB

Content Error Observed on Channel B Flag (vSS!ContentErrorB)

A content error was observed in the assigned slot on channel B.

(18) FLXAnFRMBS.CEOA

Content Error Observed on Channel A Flag (vSS!ContentErrorA)

A content error was observed in the assigned slot on channel A.

(19) FLXAnFRMBS.SEOB

Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB)

A syntax error was observed in the assigned slot on channel B.

(20) FLXAnFRMBS.SEOA

Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA)

A syntax error was observed in the assigned slot on channel A.

(21) FLXAnFRMBS.VFRB

Valid Frame Received on Channel B Flag (vSS!ValidFrameB)

A valid frame indication is set if a valid frame was received on channel B.

(22) FLXAnFRMBS.VFRA

Valid Frame Received on Channel A Flag (vSS!ValidFrameA)

A valid frame indication is set if a valid frame was received on channel A.

18.2.11.6 FLXAnFROBCM — FlexRay Output Buffer Command Mask Register

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by bits FLXAnFROBCR.OBRS[6:0].

Mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to the register internal storage when a Message RAM transfer is requested by bit FLXAnFROBCR.REQ.

When OBF Host and OBF Shadow are swapped, mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register internal storage to keep them attached to the respective Output Buffer transfer.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 18.3.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

CAUTION

After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag MBC of the selected message buffer in the FLXAnFRMBSC1 to FLXAnFRMBSC4 registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the new data flag ND of the selected message buffer in the FLXAnFRNDAT1 to FLXAnFRNDAT4 registers is cleared.

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0710_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RDSH | RHSH |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RDSS | RHSS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 18.79 FLXAnFROBCM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17 | RDSH | Read Data Section Host Flag 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer |
| 16 | RHSH | Read Header Section Host Flag 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer |
| 15 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | RDSS | Read Data Section Shadow Bit 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer |
| 0 | RHSS | Read Header Section Shadow Bit 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer |

(1) FLXAnFROBCM.RDSH

Read Data Section Host Flag

(2) FLXAnFROBCM.RHSH

Read Header Section Host Flag

(3) FLXAnFROBCM.RDSS

Read Data Section Shadow Bit

(4) FLXAnFROBCM.RHSS

Read Header Section Shadow Bit

18.2.11.7 FLXAnFROBCR — FlexRay Output Buffer Command Request Register

After setting bit FLXAnFROBCR.REQ to 1 while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1, bits FLXAnFROBCR.OBRS[6:0] is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the message buffer selected by bits FLXAnFROBCR.OBRS[6:0] from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCM.OBSYS back to 0.

By setting bit FLXAnFROBCR.VIEW to 1 while FLXAnFROBCR.OBSYS is 0, OBF Host and OBF Shadow are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register FLXAnFROBCM internal storage to keep them attached to the respective Output Buffer transfer. bits FLXAnFROBCR.OBRH[6:0] signals the number of the message buffer currently accessible by the Host.

If bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are set to 1 with the same write access while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Any write access to bits FLXAnFROBCR[15:8] while FLXAnFROBCR.OBSYS is set to 1 will cause bit FLXAnFREIR.IOBA to be set to 1. In this case, this write access has no effect and the Output Buffer will not be changed.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 18.3.12.2 (2), Data Transfer from Message RAM to Output Buffer.**

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0714_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|----|----|----|----|----|-----|------|----|-----------|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | OBRH[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OBSYS | — | — | — | — | — | REQ | VIEW | — | OBRS[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.80 FLXAnFROBCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | OBRH[6:0] | Output Buffer Request Host Flag |
| 15 | OBSYS | Output Buffer Busy Shadow Flag 0: No transfer in progress 1: Transfer between Message RAM and OBF Shadow in progress |
| 14 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | REQ | Request Message RAM Transfer Bit 0: No request 1: Transfer to OBF Shadow requested |
| 8 | VIEW | View Shadow Buffer Bit 0: No action 1: Swap OBF Shadow and OBF Host |
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | OBRS[6:0] | Output Buffer Request Shadow Bit |

(1) FLXAnFROBCR.OBRH

Output Buffer Request Host Flag

Number of message buffer currently accessible by the Host via the FLXAnFRRDHS1 to FLXAnFRRDHS3, FLXAnFRMBS, and FLXAnFRRDDS1 to FLXAnFRRDDS64 registers.

By writing bit FLXAnFROBCR.VIEW to 1 OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host.

(2) FLXAnFROBCR.OBSYS

Output Buffer Busy Shadow Flag

Set to 1 after setting bit FLXAnFROBCR.REQ. When the transfer between the Message RAM and OBF Shadow has completed, FLXAnFROBCR.OBSYS is set back to 0.

(3) FLXAnFROBCR.REQ

Request Message RAM Transfer Bit

Only writeable while bit FLXAnFROBCR.OBSYS = 0.

Requests transfer of message buffer addressed by bits FLXAnFROBCR.OBRS[6:0] from Message RAM to OBF Shadow.

(4) FLXAnFROBCR.VIEW

View Shadow Buffer Bit

Only writeable while bit FLXAnFROBCR.OBSYS = 0.

Toggles between OBF Shadow and OBF Host.

(5) FLXAnFROBCR.OBRS

Output Buffer Request Shadow Bit

Only writeable while bit FLXAnFROBCR.OBSYS = 0.

Number of source message buffer to be transferred from the Message RAM to OBF Shadow.

If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index (GIDX, see **Section 18.3.11, FIFO Function**) to OBF Shadow.

18.2.12 Data Transfer Control Register

18.2.12.1 FLXAnFRITC — FlexRay Input Transfer Configuration Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0800_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|-------|-------|----|----------|-----|-----|-----|-----|------|-----|
| | — | — | — | — | — | — | — | — | — | ITM[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | IQEIE | IQFIE | — | — | — | — | — | — | IQHR | ITE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R/W | R/W |

Table 18.81 FLXAnFRITC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | ITM[6:0] | Input queue Table Max Bit These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue. |
| 15 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | IQEIE | Input Queue Empty Interrupt Enable Bit 0: Disabled 1: Enabled |
| 8 | IQFIE | Input Queue Full Interrupt Enable Bit 0: Disabled 1: Enabled |
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | IQHR | Input Queue Halt Request Bit 0: Input queue run request 1: Input queue halt request |
| 0 | ITE | Input Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request |

(1) FLXAnFRITC.ITM

Input queue Table Max Bit

The user can only write to this bit when bit FLXAnFRITC.ITE is 0.

These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue.

Valid values are 00_H (1 queue entry) to 7F_H (128 queue entries).

Note that each entry requires two long words in the input pointer table.

(2) FLXAnFRITC.IQEIE

Input Queue Empty Interrupt Enable Bit

This bit controls the input queue empty interrupt.

0: Disabled

No interrupt will be generated and the input queue empty interrupt line will be released.

1: Enabled

Input queue empty interrupt will be generated when bit FLXAnFRITS.IQEIS is 1.

(3) FLXAnFRITC.IQFIE

Input Queue Full Interrupt Enable Bit

This bit controls the input queue full interrupt.

0: Disabled

No interrupt will be generated and the input queue full interrupt line will be released.

1: Enabled

Input queue full interrupt will be generated when bit FLXAnFRITS.IQFIS is 1.

(4) FLXAnFRITC.IQHR

Input Queue Halt Request Bit

The IQHR bit should not be set to 1 when FLXAnFRITS.ITS is 0.

This bit requests a halt of the input queue.

The status of the halt request is shown in the FLXAnFRITS.IQH register.

Refer to **Section 18.3.16.1 (5), Halting the input queue** about usage of this bit.

0: Input queue run request

The input queue resumes their operation.

1: Input queue halt request

The input queue gets halted. An active input transfer will be completed but no further transfer request will start.

(5) FLXAnFRITC.ITE

Input Transfer Enable Bit

The user should only set this bit to 1 when bit FLXAnFRIBCR.IBSYS is 0.

The user should only set this bit to 0 when bit FLXAnFRITC.IQHR is 0. Otherwise committed input transfers get lost.

This bit controls the operation mode of the input transfer queue.

The operation status of the input transfer queue function is shown in bit FLXAnFRITS.ITS.

Refer to **Section 18.3.16.1 (1), Activation and deactivation** about usage of this bit.

0: Operation Disable request

The input transfer queue is disabled when it becomes empty.

1: Operation Enable request

The input transfer queue is enabled. Input data structures are transferred to the FlexRay internal message RAM.

18.2.12.2 FLXAnFROTC — FlexRay Output Transfer Configuration Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0804_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|------|------|-----|-----|----|----|----|----------|-----|-----|------|-----|
| | — | — | — | — | — | — | — | — | — | — | — | FTM[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | FWIE | OWIE | FIE | OIE | — | — | — | — | — | — | OTCS | OTE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R/W | R/W |

Table 18.82 FLXAnFROTC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 16 | FTM[4:0] | FIFO Table Max Bit Configures the number of FIFO entries the output transfer handler is capable to maintain in the Local RAM/Global RAM. |
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 | FWIE | FIFO transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled |
| 10 | OWIE | Output transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled |
| 9 | FIE | FIFO transfer Interrupt Enable Bit 0: Disabled 1: Enabled |
| 8 | OIE | Output transfer Interrupt Enable Bit 0: Disabled 1: Enabled |
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | OTCS | Output Transfer Condition Select Bit 0: New data only mode 1: New data and status changed mode |
| 0 | OTE | Output Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request |

(1) FLXAnFROTC.FTM

FIFO Table Max Bit

The user can only write to these bits when bit FLXAnFROTS.OTS is 0.

Configures the number of FIFO entries the output transfer handler is capable to maintain in the Local RAM/Global RAM.

Valid values are 00_H (1 FIFO entry) to 1F_H (32 FIFO entries).

(2) FLXAnFROTC.FWIE

FIFO transfer Warning Interrupt Enable Bit

This bit controls the FIFO transfer warning interrupt.

0: Disabled

No interrupt will be generated and the FIFO transfer warning interrupt line will be released.

1: Enabled

FIFO transfer warning interrupt will be generated when bit FLXAnFROTS.FWIS is 1.

(3) FLXAnFROTC.OWIE

Output transfer Warning Interrupt Enable Bit

This bit controls the output transfer warning interrupt.

0: Disabled

No interrupt will be generated and the output transfer warning interrupt line will be released.

1: Enabled

Output transfer warning interrupt will be generated when bit FLXAnFROTS.OWIS is 1.

(4) FLXAnFROTC.FIE

FIFO transfer Interrupt Enable Bit

This bit controls the FIFO transfer interrupt.

0: Disabled

No interrupt will be generated and the FIFO transfer interrupt line will be released.

1: Enabled

FIFO transfer interrupt will be generated when bit FLXAnFROTS.FIS is 1.

(5) FLXAnFROTC.OIE

Output transfer Interrupt Enable Bit

This bit controls the output transfer interrupt.

0: Disabled

No interrupt will be generated and the output transfer interrupt line will be released.

1: Enabled

Output transfer interrupt will be generated when bit FLXAnFROTS.OTIS is 1.

(6) FLXAnFROTC.OTCS

Output Transfer Condition Select Bit

The user can only write to this bit when bit FLXAnFROTS.OTS is 0.

This bit controls the output transfer condition.

0: New data only mode

Bits FLXAnFRNDATm.NDp are used to detect a transfer condition for dedicated receive buffer.

1: New data and status changed mode

Bits FLXAnFRNDATm.NDp and FLXAnFRMBSCm. are used to detect a transfer condition for dedicated transmit and receive buffer.

(7) FLXAnFROTC.OTE

Output Transfer Enable Bit

The user should only set this bit to 1 when bit FLXAnFROBCR.OBSYS is 0.

This bit controls the operation mode of the output transfer function.

The operation status of the output buffer transfer function is shown in FLXAnFROTS.OTS.

Refer to **18.3.16.2, (1) Activation and deactivation** about usage of this bit.

0: Operation Disable request

The output buffer transfer gets disabled.

An active message buffer transfer will be completed but no further transfer will start.

1: Operation Enable request

The output buffer transfer gets enabled. Message buffers are transferred from the FlexRay internal message RAM to output data structures.

The user should not change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

18.2.12.3 FLXAnFRIBA — FlexRay Input Pointer Table Base Address Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0808_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | ITA[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ITA[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R |

Table 18.83 FLXAnFRIBA Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | ITA[31:0] | Input Table Base Address Bit These bits configure the base address of the input pointer table. |

(1) FLXAnFRIBA.ITA

Input Table Base Address Bit

The user can only write to this bit when FLXAnFRITS.ITS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFRIBA.ITA[1:0] are always 0.

These bits configure the base address of the input pointer table.

The table is used for the input transfer queue transferring message buffers from the Local RAM/Global RAM into the FlexRay internal message RAM.

The size of the input queue is configured in bits FLXAnFRITC.ITM[6:0].

Note that each entry requires two long words in the input pointer table.

18.2.12.4 FLXAnFRFBA — FlexRay FIFO Pointer Table Base Address Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 080C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | FTA[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | FTA[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R |

Table 18.84 FLXAnFRFBA Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | FTA[31:0] | FIFO pointer Table Base Address Bit These bits configure the base address of the FIFO pointer table. |

(1) FLXAnFRFBA.FTA

FIFO pointer Table Base Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFRFBA.FTA[1:0] are always 0.

These bits configure the base address of the FIFO pointer table.

The table is used for message buffers transferred from the FlexRay internal FIFO to the Local RAM/ Global RAM.

The size of the FIFO is configured in bits FLXAnFROTC.FTM[4:0].

18.2.12.5 FLXAnFROBA — FlexRay Output Pointer Table Base Address Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0810_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | OTA[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OTA[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R |

Table 18.85 FLXAnFROBA Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | OTA[31:0] | Output pointer Table Base Address Bit These bits configure the base address of the output pointer table. |

(1) FLXAnFROBA.OTA

Output pointer Table Base Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFROBA.OTA[1:0] are always 0.

These bits configure the base address of the output pointer table.

The table is used for message buffers transferred from the FlexRay internal message RAM to the Local RAM/Global RAM.

The size of the table depends on the utilization of the FlexRay internal message RAM and can have up to 128 entries.

18.2.12.6 FLXAnFRIQC — FlexRay Input Queue Control Register

Access: This register can only be written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0814_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | IMBNR[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W |

Table 18.86 FLXAnFRIQC Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 7 | Reserved | When writing, write the value after reset. |
| 6 to 0 | IMBNR[6:0] | Input Message Buffer Number Bit Message buffer number added to the input queue |

(1) FLXAnFRIQC.IMBNR

Input Message Buffer Number Bit

The user can only write to this bit when FLXAnFRITS.IQFP is 0.

The user should not write to this register when FLXAnFRITS.ITS is 0 or when FLXAnFRITC.ITE is 0.

These bits are read as 0.

This value specifies the message buffer added to the input queue.

The number has to be identical to FLXAnFRWRHS4.IMBNR[6:0] (see **Section 18.3.16.1 (3), Input pointer table**) of the input pointer table.

The address to the input data structure has to be provided in the input pointer table at the put index (bit FLXAnFRITS.IPIDX[6:0]) before writing to this register.

Writing to this register increments the input put index (bit FLXAnFRITS.IPIDX[6:0]).

18.2.12.7 FLXAnFRUIR — FlexRay User Input Transfer Request Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0818_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | UIDX[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.87 FLXAnFRUIR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 to 0 | UIDX[7:0] | User requested Input inDeX Bit Input pointer table index requested for input transfer |

(1) FLXAnFRUIR.UIDX

User requested Input inDeX Bit

The user can only write to this bit when bit FLXAnFRITS.UIRP is 0.

The user should not write to this register when bit FLXAnFRITS.ITS is 0.

The user should not write to this register when bit FLXAnFRITS.UIRP is 1.

The user should not write to this register when bit FLXAnFRITS.IQH is 1.

The user should only write bits FLXAnFRITC.ITM[6:0] +1 to this register.

This value specifies the input pointer table index for the requested input transfer.

The address to the input data structure has to be provided in the input pointer table at the index FLXAnFRUIR.UIDX[7:0] before writing to this register.

When writing to this register, the requested input data structure will be transferred from input data structure position to the FlexRay internal message RAM.

In opposite to queued input transfers the related FLXAnFRDA.DA flag is not influenced by the user input transfer.

18.2.12.8 FLXAnFRUOR — FlexRay User Output Transfer Request Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 081C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|------|----|----|------------|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | URDS | — | — | UMBNR[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.88 FLXAnFRUOR Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | URDS | User request Read Data Section Bit 0: Data section is not transferred 1: Data section is transferred |
| 8, 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | UMBNR[6:0] | User requested output Message Buffer Number Bit Message buffer number requested for output transfer |

(1) FLXAnFRUOR.URDS

User request Read Data Section Bit

The user can only write to this bit when bit FLXAnFROTS.UORP is 0.

The user should not write to this register when bit FLXAnFROTS.OTS is 0.

The user should not write to this register when bit FLXAnFROTS.UORP is 1.

0: Data section is not transferred

The data section of the message buffer selected by the bits FLXAnFRUOR.UMBNR[6:0] is not requested

1: Data section is transferred

The data section of the message buffer selected by the bits FLXAnFRUOR.UMBNR[6:0] is requested

(2) FLXAnFRUOR.UMBNR

User requested output Message Buffer Number Bit

The user can only write to this bit when bit FLXAnFROTS.UORP is 0.

The user should not write to this register when bit FLXAnFROTS.OTS is 0.

The user should not write to this register when bit FLXAnFROTS.UORP is 1.

The user should restrict this bit to dedicated receive and transmit buffers when the FlexRay module is not in the CONFIG state.

When writing to this register, the contents of the message buffer will be transferred to a given area. The destination will be the area specified by the value in the output pointer table.

The data to be transferred can be specified by setting the URDS bit whether to transfer the header section only or to include the data section.

18.2.12.9 FLXAnFRAHBC — FlexRay H-Bus Configuration Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0840_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|------------|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | HPROT[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |

Table 18.89 FLXAnFRAHBC Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | HPROT[3:0] | Protection Control Bit |

(1) FLXAnFRAHBC.HPROT

Protection Control Bit

The user should only write to this register only when bit FLXAnFRITS.ITS is 0 and bit FLXAnFROTS.OTS is 0.

This register configures the value assigned to the H-Bus protection control signal.

18.2.13 Data Transfer Status Register

18.2.13.1 FLXAnFRITS — FlexRay Input Transfer Status Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0820_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|------------|----|------|----|----|-------|-------|------------|----|----|----|----|------|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | IGIDX[6:0] | | | | | | — | IPIDX[6:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | IQFP | — | — | IQEIS | IQFIS | — | — | — | — | — | UIRP | IQH | ITS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R | R |

Table 18.90 FLXAnFRITS Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 30 to 24 | IGIDX[6:0] | Input queue Get InDeX Bit Represents the get index of the input pointer table |
| 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | IPIDX[6:0] | Input queue Put InDeX Bit Represents the put index of the input pointer table |
| 15 to 13 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 12 | IQFP | Input Queue Full condition Pending Bit 0: Entries in the input queue are available 1: All entries in the input queue are occupied |
| 11, 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | IQEIS | Input Queue Empty Interrupt Status Bit 0: No input queue empty condition detected 1: Input queue empty condition detected |
| 8 | IQFIS | Input Queue Full Interrupt Status Bit 0: No input queue full condition detected 1: Input queue full condition detected |
| 7 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | UIRP | User Input transfer Request Pending Bit 0: No user input transfer request pending 1: User input transfer request pending |
| 1 | IQH | Input Queue Halted Bit 0: Input queue not halted 1: Input queue halted |
| 0 | ITS | Input Transfer Status Bit 0: Disabled 1: Enabled |

(1) FLXAnFRITS.IGIDX

Input queue Get InDeX Bit

These bits are only valid when bit FLXAnFRITS.IQH is 1

These bits represent the input pointer index the input queue handler will transfer next.

Valid values are 00_H to FLXAnFRITC.ITM.

The get index is incremented when the input data structure has been transferred from the Local RAM/ Global RAM and the related FLXAnFRDA.DA flag is cleared.

The index is set to 00_H when bit FLXAnFRITS.ITS changes from 0 to 1.

(2) FLXAnFRITS.IPIDX

Input queue Put InDeX Bit

These bits represent the index where the next input data structure pointer in the input pointer table should be stored.

Valid values are 00_H to FLXAnFRITC.ITM.

After reaching the maximum value the put index continues from 00_H.

The index is incremented when writing to bits FLXAnFRIQC.IMBNR[6:0].

The index is set to 00_H when bit FLXAnFRITS.ITS changes from 0 to 1.

(3) FLXAnFRITS.IQFP

Input Queue Full condition Pending Bit

This bit represents that the input queue is full.

There should be no further input transfer requests, by writing to bits FLXAnFRIQC.IMBNR[6:0], as long as this bit is 1.

[Clearing condition]

- This bit is cleared when there is one free entry in the input queue.

[Setting condition]

- This bit is set when all entries in the input queue are occupied.

(4) FLXAnFRITS.IQEIS

Input Queue Empty Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFRITC.IQEIE, the input queue empty interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRITS.IQEIS.
- This bit is cleared when bit FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]

- This bit is set when all pending input transfers have been processed and consequently the input queue becomes empty.

(5) FLXAnFRITS.IQFIS

Input Queue Full Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFRITC.IQFIE the input queue full interrupt is generated when FLXAnFRITS.IQFIS is 1.

This flag is intended as interrupt status flag. It does not represent the current input queue status; for this status refer to bit FLXAnFRITS.IQFP.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRITS.IQFIS.
- This bit is cleared when bit FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]

- This bit is set when all entries in the input queue are occupied.

(6) FLXAnFRITS.UIRP

User Input transfer Request Pending Bit

This bit represents that a user input transfer is still pending.

There should be no further write access to bits FLXAnFRUIR.UIDX[7:0] when this bit is 1.

[Clearing condition]

- This bit is cleared when the user input transfer request is processed by the input transfer handler.

[Setting condition]

- This bit is set when writing to bits FLXAnFRUIR.UIDX[7:0].

(7) FLXAnFRITS.IQH

Input Queue Halted Bit

This bit represents the status of the input queue.

There should be no further write access to bits FLXAnFRUIR.UIDX[7:0] when this bit is 1.

[Clearing condition]

- This bit is cleared when bit FLXAnFRITC.IQHR is set to 0.

[Setting condition]

- This bit is set immediately when bit FLXAnFRITC.IQHR is set to 1 and there is no ongoing input transfer.
- This bit is set only after an ongoing input transfer has been completed and bit FLXAnFRITC.IQHR is set to 1.

(8) FLXAnFRITS.ITS

Input Transfer Status Bit

This bit represents the status of the input queue handler.

While this bit is 1, there should be no read or write access to the address area $\langle \text{FLXAn_base} \rangle + 0400_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 05FF_{\text{H}}$ and there should be no CLEAR_RAM command applied to bits FLXAnFRSUCC1.CMD[3:0].

The input transfer queue indexes and related status flags are set to 0 when bit FLXAnFRITS.ITS changes from 0 to 1.

[Clearing condition]

- This bit is cleared immediately when bit FLXAnFRITC.ITE is set to 0 and there are no pending input transfers
- This bit is cleared after all pending requests have been processed and bit FLXAnFRITC.ITE is 0.

[Setting condition]

- This bit is set when bit FLXAnFRITC.ITE is set to 1.

18.2.13.2 FLXAnFROTS — FlexRay Output Transfer Status Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0824_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----|-----|----------|----|-----|-----|-----|------|----|----|------------|----|----|------|----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | FFL[5:0] | | | | | — | — | — | FGIDX[4:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | FWP | OWP | FDA | — | FWS | OWS | FIS | OTIS | — | — | — | — | — | UORP | — | OTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

Table 18.91 FLXAnFROTS Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 to 24 | FFL[5:0] | FIFO Fill Level Bit Represent the number of unprocessed output FIFO structures |
| 23 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 16 | FGIDX[4:0] | FIFO Get InDeX Bit Represent the get index in the FIFO pointer table |
| 15 | FWP | FIFO transfer Warning condition Pending Bit 0: No FIFO transfer warning condition pending 1: FIFO transfer warning condition pending |
| 14 | OWP | Output transfer Warning condition Pending 0: No output transfer warning condition pending 1: Output transfer warning condition pending |
| 13 | FDA | FIFO Data Available Bit 0: No available FIFO structures 1: FIFO structures available at current FLXAnFROTS.FGIDX index |
| 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 | FWIS | FIFO transfer Warning Interrupt Status Bit 0: No FIFO transfer warning condition detected 1: FIFO transfer warning condition detected |
| 10 | OWIS | Output transfer Warning Interrupt Status Bit 0: No output transfer warning condition detected 1: Output transfer warning condition detected |
| 9 | FIS | FIFO transfer Interrupt Status Bit 0: No FIFO structure updated in Local RAM/Global RAM 1: FIFO structure updated in Local RAM/Global RAM |
| 8 | OTIS | Output transfer Interrupt Status Bit 0: No output structure updated in Local RAM/Global RAM 1: Output structure updated in Local RAM/Global RAM |
| 7 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | UORP | User Output transfer Request Pending Bit 0: No user output transfer request pending 1: User output transfer request pending |

Table 18.91 FLXAnFROTS Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | OTS | Output Transfer Status Bit 0: Disabled 1: Enabled |

(1) FLXAnFROTS.FFL

FIFO Fill Level Bit

These bits represent the number of available output FIFO structures in the Local RAM/Global RAM.

Valid values are 00_{H} to $\text{FLXAnFROTC.FTM}+1$.

The value 00_{H} represents that the FIFO is empty.

The value $\text{FLXAnFROTC.FTM}+1$ represents that the FIFO is full and no further FIFO transfers will be done.

The FIFO fill level is incremented when a FIFO data structure has been transferred from the FlexRay internal FIFO into the Local RAM/Global RAM.

The FIFO fill level is decremented when the user releases a FIFO data structure in the Local RAM/Global RAM by writing 1 to bit FLXAnFROTS.FDA .

The FIFO fill level is set to 00_{H} when bit FLXAnFROTS.OTS changes from 0 to 1.

(2) FLXAnFROTS.FGIDX

FIFO Get InDeX Bit

These bits represent the index where the current output data structure pointer in the FIFO pointer table is available for reading.

Valid values are 00_{H} to FLXAnFROTC.FTM .

After reaching the maximum value the get index continues from 00_{H} .

The index is incremented when a FIFO data structure is released by writing 1 to bit FLXAnFROTS.FDA .

The index is set to 00_{H} when bit FLXAnFROTS.OTS changes from 0 to 1.

(3) FLXAnFROTS.FWP

FIFO transfer Warning condition Pending Bit

This bit represents the FIFO transfer warning condition.

[Clearing condition]

- This bit is cleared when there are free output data structures ($\text{FLXAnFROTS.FFL} \leq \text{FLXAnFROTC.FTM}$).

This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures ($\text{FLXAnFROTS.FFL} = \text{FLXAnFROTC.FTM}+1$).

(4) FLXAnFROTS.OWP

Output transfer Warning condition Pending Bit

This bit represents the output transfer warning condition.

[Clearing condition]

- This bit is cleared, when all output structure pointers that have a pending output handler transfer condition detected, are released (for dedicated transmit and receive message buffers or a user output transfer request).
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).
- This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(5) FLXAnFROTS.FDA

FIFO Data Available Bit

Writing 0 has no effect on the bit value.

When this bit is 1, the next valid output data structure is available.

The related data structure pointer is in the FIFO pointer table at FLXAnFROTS.FGIDX.

Writing 1 to FLXAnFROTS.FDA

- increments FLXAnFROTS.FGIDX[4:0] and
- decrements the FIFO fill level (FLXAnFROTS.FFL)

If there are still unprocessed data structures FLXAnFROTS.FDA remains 1.

[Clearing condition]

- This bit is cleared when writing 1 to FLXAnFROTS.FDA and the FIFO fill level becomes 00_H.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when there is at least one FIFO data structure available in the Local RAM/Global RAM.

(6) FLXAnFROTS.FWIS

FIFO transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.FWIE, the FIFO transfer warning interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFROTS.FWIS.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM+1).

(7) FLXAnFROTS.OWIS

Output transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.OWIE, the FIFO transfer warning interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to this bit.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).
- This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(8) FLXAnFROTS.FIS

FIFO transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.FIE, the FIFO transfer interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to this bit.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when a FIFO data structure is updated by the transfer handler or bits FLXAnFROTS.FFL[5:0] changes from 00_H to 01_H.

(9) FLXAnFROTS.OTIS

Output transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.OIE, the output transfer interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to bit FLXAnFROTS.OTIS.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when an output data structure is updated by the transfer handler (from a dedicated transmit or receive message buffer or by a user output transfer request).

(10) FLXAnFROTS.UORP

User Output transfer Request Pending Bit

This bit represents that a user output transfer is still pending.

There should be no further write access to bits FLXAnFRUOR.UMBNR[6:0] when this bit is 1.

[Clearing condition]

- This bit is cleared when the user output transfer request is processed by the output transfer handler.
- This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when writing to bit FLXAnFRUOR.UMBNR.

(11) FLXAnFROTS.OTS

Output Transfer Status Bit

This bit represents the status of the output transfer handler.

While this bit is 1, there should be no read or write access to the address area $\langle \text{FLXAn_base} \rangle + 0600_{\text{H}}$ to $\langle \text{FLXAn_base} \rangle + 07FF_{\text{H}}$ and there should be no CLEAR_RAM command applied to bits FLXAnFRSUCC1.CMD[3:0].

While this bit is 1, the user should not change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

The output handler transfer indexes and related status flags are set to 0 when this bit changes from 0 to 1.

[Clearing condition]

- This bit is cleared immediately when bit FLXAnFROTC.OTE is set to 0 and there are no ongoing output transfers.
- This bit is cleared after an ongoing transfer has been completed and bit FLXAnFROTC.OTE is 0.

[Setting condition]

- This bit is set when bit FLXAnFROTC.OTE is set to 1.

18.2.13.3 FLXAnFRAES — FlexRay Access Error Status Register

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0828_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|-----|-----|-----|-----|-----------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | MAE | FAE | OAE | IAE | EIDX[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

Table 18.92 FLXAnFRAES Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 | MAE | Multiple Access Errors Bit 0: No multiple access errors occurred 1: Multiple access errors occurred |
| 10 | FAE | FIFO transfer Access Error Bit 0: No access error occurred during FIFO transfer 1: Access error occurred during FIFO transfer |
| 9 | OAE | Output transfer Access Error Bit 0: No access error occurred during output transfer 1: Access error occurred during output transfer |
| 8 | IAE | Input transfer Access Error Bit 0: No access error occurred during input transfer 1: Access error occurred during input transfer |
| 7 to 0 | EIDX[7:0] | Error InDeX Bit Data structure pointer index number |

(1) FLXAnFRAES.MAE

Multiple Access Errors Bit

Writing 0 has no effect on the bit value.

This bit represents that there were multiple access errors during a data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.MAE.

[Setting condition]

- This bit is set when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is set and one of the conditions below is met:
 - an access to an protected address occurred during a FIFO data transfer or
 - an access to an protected address occurred during an output data transfer or
 - an access to an protected address occurred during an input data transfer

(2) FLXAnFRAES.FAE

FIFO transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit represent that there was an access error during a FIFO data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.FAE.

[Setting condition]

- This bit is set when a Local RAM/Global RAM access error was detected during a FIFO transfer and bits FLXAnFRAES.OAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(3) FLXAnFRAES.OAE

Output transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit represent that there was an access error during a output data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.OAE.

[Setting condition]

- This bit is set when a Local RAM/Global RAM access error was detected during an output transfer and bits FLXAnFRAES.FAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(4) FLXAnFRAES.IAE

Input transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit represent that there was an access error during an input data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.IAE.

[Setting condition]

- This bit is set when a Local RAM/Global RAM access error was detected during an output transfer and bits FLXAnFRAES.OAE, FLXAnFRAES.FAE and FLXAnFRAES.MAE are 0.

(5) FLXAnFRAES.EIDX

Error InDeX Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is 1.

When bit FLXAnFRAES.FAE is 1, FLXAnFRAES.EIDX holds the used FIFO put index when the access error has occurred.

When bit FLXAnFRAES.OAE is 1, FLXAnFRAES.EIDX holds the input pointer table get index used when an access error occurred during an input transfer or when the user request an input transfer.

When bit FLXAnFRAES.IAE is 1, FLXAnFRAES.EIDX holds the used input pointer table get index when the access error has occurred.

These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is changing from 0 to 1.

18.2.13.4 FLXAnFRAEA — FlexRay Access Error Address Register

Access: This register can only be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 082C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | AEA[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | AEA[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 18.93 FLXAnFRAEA Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | AEA[31:0] | Access Error Address Bit Address in the Local RAM/Global RAM when an access error has occurred |

(1) FLXAnFRAEA.AEA

Access Error Address Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is 1.

These bits represent the address of the access error indicated in the FLXAnFRAES register.

These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is changing from 0 to 1.

18.2.13.5 FLXAnFRDAm — FlexRay message Data Available Register m (m = 0 to 3)

Access: This register can be read/written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0830_H to <FLXAn_base> + 083C_H (<FLXAn_base> + 0830_H + m × 4)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | DAP[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DAP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 18.94 FLXAnFRDAm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 0 | DAP[31:0] | Data Available Bit p 0: No data available for destination 1: Data available for destination |

(1) FLXAnFRDAm.DAp (p = m × 32 to ((m + 1) × 32) – 1)

Data Available Bit p

The user should not write a 1 to bits that are 0.

To maintain the status of input transfers, the user should not clear bits related to input transfers.

This register is used for input and output transfers.

Each flag corresponds to a FlexRay message buffer.

[Clearing condition]

- Input transfer:
This bit is cleared when the input data structure has been transferred from the Local RAM/Global RAM. The data structure and the data structure pointer can be changed when the related flag is 0.
- Output transfer:
This bit is cleared when writing a 1 to it.

[Setting condition]

- Input transfer:
This bit is set when the corresponding message buffer number has been written to bits FLXAnFRIQC.IMBNR[6:0].
As long as this bit is 1, the input data structure and the data structure pointer corresponding to this input transfer request should not be changed.
- Output transfer:
This bit is set when the output data structure corresponding to this message buffer has been updated.

As long as this bit is 1, the data structure is stable; no further update of the data structure will be done by the output handler. While this bit is 1, the application is allowed to change the output data structure pointer in the output pointer table for this message buffer number.

18.3 Functional Description

This chapter describes the FlexRay implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification.

18.3.1 FlexRay Module Operation Control

18.3.1.1 FlexRay Module Enable

After a reset or after the FlexRay module has been disabled (following **Section 18.3.1.2, FlexRay Module Disable**) the FlexRay module is in the reset state (bit FLXAnFROS.OS is 0) and the clocks of the FlexRay core module are disabled.

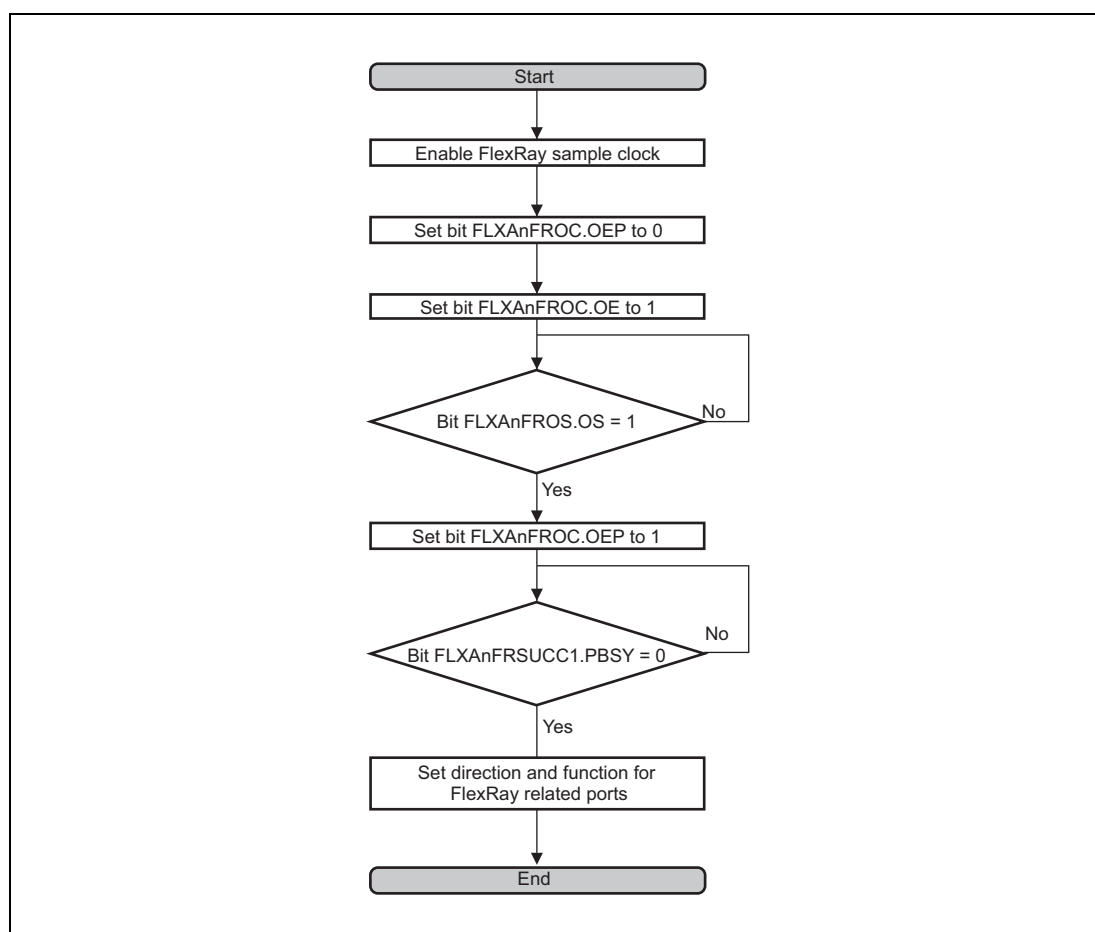


Figure 18.3 FlexRay Enable Processing Flow

18.3.1.2 FlexRay Module Disable

The FlexRay module can be disabled at any time. However, it is recommended to disable the FlexRay module using bit FLXAnFROC.OE only when the FlexRay module is in HALT, CONFIG or DEFAULT_CONFIG state. Resetting the FlexRay module in any other state will terminate any ongoing FlexRay communication.

If the data transfer function is used, it is also required to disable this function before disabling the FlexRay module (see **Section 18.3.16.1, Input Data Transfer (1), Activation and deactivation** for suspending input transfer function and **Section 18.3.16.2, Output Data Transfer (1), Activation and deactivation** for suspending output transfer).

The following flow should be executed to disable the FlexRay module.

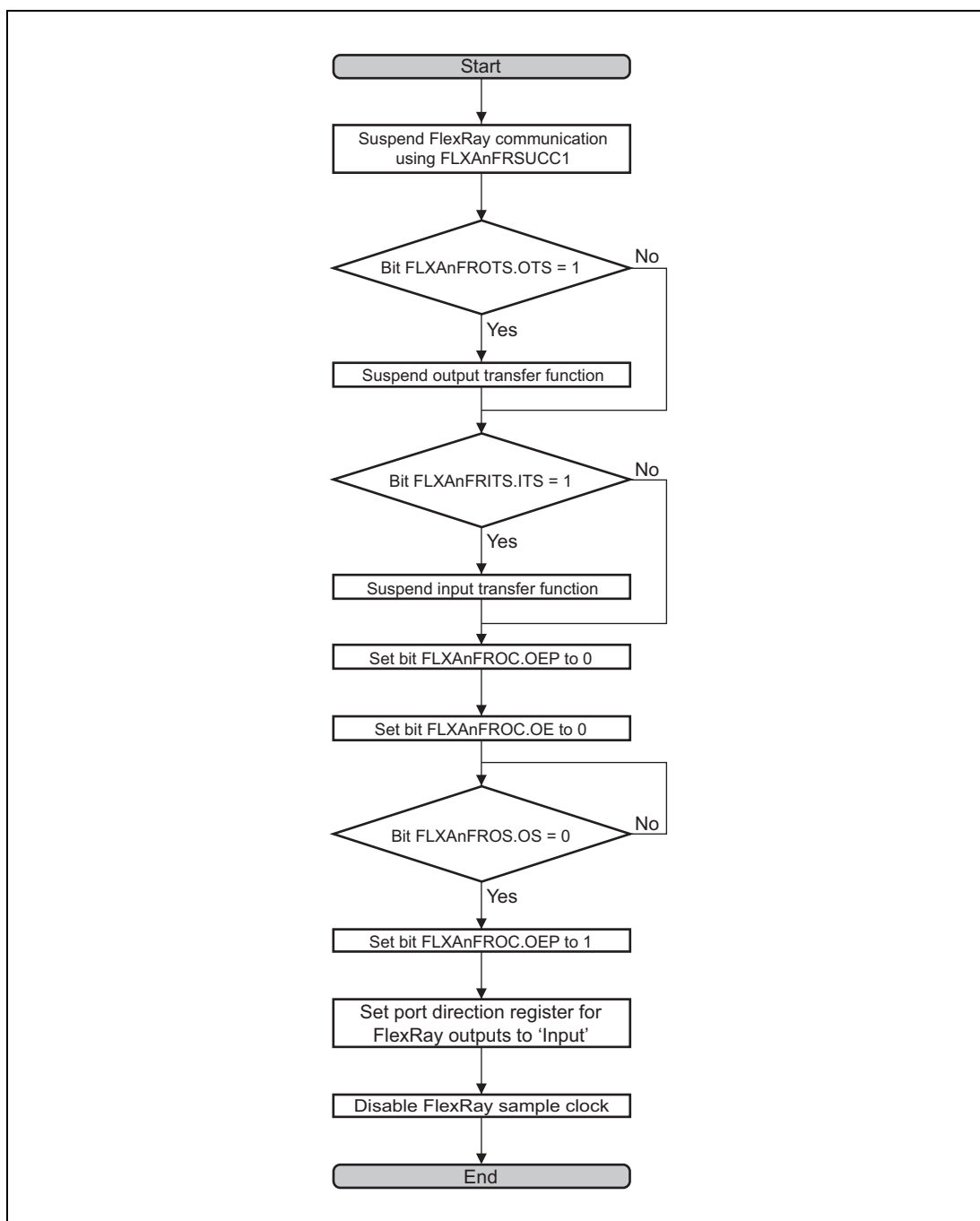


Figure 18.4 FlexRay Disable Processing Flow

18.3.2 Communication Cycle

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

A FlexRay communication cycle consists of the following elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

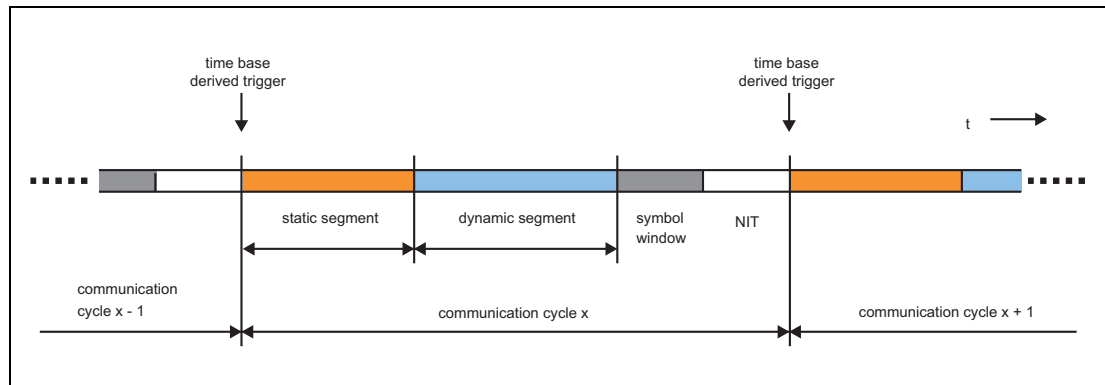


Figure 18.5 Structure of Communication Cycle

18.3.2.1 Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

Parameters:

Number of Static Slots (FLXAnFRGTUC7.NSS[9:0])

Static Slot Length (FLXAnFRGTUC7.SSL[9:0])

Payload Length Static (FLXAnFRMHDC.SFDL[6:0])

Action Point Offset (FLXAnFRGTUC9.APO[5:0])

18.3.2.2 Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters:

Number of Minislots (FRGTUC8.NMS[12:0])

Minislot Length (FRGTUC8.MSL[5:0])

Minislot Action Point Offset (FRGTUC9.MAPO[4:0])

Start of Latest Transmit (last minislot) (FRMHDC.SLT[12:0])

18.3.2.3 Symbol Window

During the symbol window only one media access test symbol (MTS) may be transmitted per channel. MTS symbols are sent in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters:

Symbol Window Action Point Offset (FRGTUC9.APO[4:0]) (same as for static slots)

Network Idle Time Start (FRGTUC4.NIT[13:0])

18.3.2.4 Network Idle Time (NIT)

During network idle time the CC has to perform the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks after offset correction start
- Perform cluster cycle related tasks

Parameters:

Network Idle Time Start (FLXAnFRGTUC4.NIT[13:0])

Offset Correction Start (FLXAnFRGTUC4.OCS[13:0])

18.3.2.5 Configuration of NIT Start and Offset Correction Start

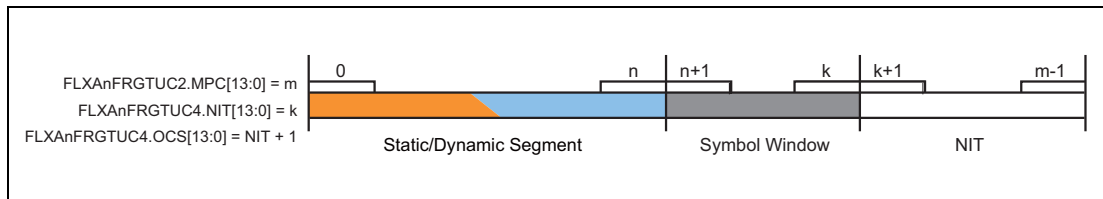


Figure 18.6 Configuration of NIT Start and Offset Correction Start

The number of macroticks per cycle $gMacroPerCycle$ is assumed to be m . It is configured by programming $FLXAnFRGTUC2.MPC[13:0] = m$.

The static / dynamic segment starts with macrotick 0 and ends with macrotick n :

$$n = \text{static segment length} + \text{dynamic segment offset} + \text{dynamic segment length} - 1MT$$

$$= gNumberOfStaticSlots \times gdStaticSlot + \text{dynamic segment offset} + gNumberOfMinislots \times gdMinislot - 1 MT$$

The static segment length is configured by $FLXAnFRGTUC7.SSL[9:0]$ and $FLXAnFRGTUC7.NSS[9:0]$.

The dynamic segment length is configured by $FLXAnFRGTUC8.MSL[5:0]$ and $FLXAnFRGTUC8.NMS[12:0]$.

The dynamic segment offset is:

If $gdActionPointOffset \leq gdMinislotActionPointOffset$:

$$\text{dynamic segment offset} = 0 MT$$

Else if $gdActionPointOffset > gdMinislotActionPointOffset$:

$$\text{dynamic segment offset} = gdActionPointOffset - gdMinislotActionPointOffset$$

The NIT starts with macrotick $k+1$ and ends with the last macrotick of cycle $m-1$. It has to be configured by setting $FLXAnFRGTUC4.NIT[13:0] = k$.

For the FlexRay module the offset correction start is required to be $FLXAnFRGTUC4.OCS[13:0] \geq FLXAnFRGTUC4.NIT[13:0] + 1 = k + 1$.

The length of symbol window results from the number of macroticks between the end of the static / dynamic segment and the beginning of the NIT. It can be calculated by the number of macroticks ($k - n$).

18.3.3 Communication Modes

The FlexRay Protocol Specification defines the Time-Triggered Distributed (TT-D) mode.

18.3.3.1 Time-triggered Distributed (TT-D)

In TT-D mode the following configurations are possible:

- Pure static: Minimum 2 static slots + symbol window (optional)
- Mixed static/dynamic: Minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes needs to be configured for distributed time-triggered operation.

Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

18.3.4 Clock Synchronization

In TT-D mode a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

18.3.4.1 Global Time

Activities in a FlexRay node, individual nodes independently operate. Operations including communication, are based on the concept of a global time. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values; the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotock (MT) = basic unit of time measurement in a FlexRay network, a macrotock consists of an integer number of microticks (μT)
- Cycle length = duration of a communication cycle in units of macroticks (MT)

18.3.4.2 Local Time

Internally, nodes time their behavior with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick (μT).

Node specific:

- Oscillator clock \rightarrow prescaler \rightarrow microtick (μT)
- μT = basic unit of time measurement in a CC, clock correction is done in units of μT s
- Cycle counter + macrotick counter = nodes local view of the global time

18.3.4.3 Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (FLXAnFRGTUC2.SNM[3:0]) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment is measured. In a two channel cluster the sync node has to be configured to send sync frames on both channels. The calculation of correction terms is done during NIT (offset: every cycle, rate: every odd cycle) by using an FTM algorithm. For details see *FlexRay protocol specification v2.1, chapter 8*.

(1) Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during NIT of every communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values
- Correction value is a signed integer number of μ Ts
- Correction done in odd numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened / shortened)

(2) Rate (frequency) Correction

- Pairs of deviation values measured and stored in even / odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during NIT of odd numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of μ Ts
- Distributed over macroticks comprising the next even / odd cycle pair (MTs lengthened / shortened)

(3) Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case bit FLXAnFRMRC.SPLM has to be programmed to 1.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames, bit FLXAnFRSUCC1.TXSY must be set to 1.

(4) External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset / rate correction value is a signed integer
- External offset / rate correction value is added to calculated offset / rate correction value
- Aggregated offset / rate correction term (external + internal) is not checked against configured limits

18.3.5 Error Handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set FLXAnFREIR.PEMC to 1 and may trigger an interrupt to the Host if enabled. The actual error mode is signaled by FLXAnFRCCCEV.ERRM[1:0].

Table 18.95 Error Modes of the POC (Degradation Model)

| Error Mode | Activity |
|------------|--|
| ACTIVE | Full operation, State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR. |
| PASSIVE | Reduced operation, State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR. |
| COMM_HALT | Operation halted, State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR. The bus drivers are disabled. |

18.3.5.1 Clock Correction Failed Counter

When the Clock Correction Failed Counter reaches the “maximum without clock correction passive” limit defined by FLXAnFRSUCC3.WCP[3:0], the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the “maximum without clock correction fatal” limit defined by FLXAnFRSUCC3.WCF[3:0], it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The Clock Correction Failed Counter (FLXAnFRCCCEV.CCFC[3:0]) allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the CC passed protocol startup phase. It will be incremented by one at the end of any odd communication cycle during which either the missing offset correction FLXAnFRSFS.MOCS or the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter is reset to zero at the end of an odd communication cycle if neither the missing offset correction FLXAnFRSFS.MOCS nor the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter stops incrementing when the “maximum without clock correction fatal” value FLXAnFRSUCC3.WCF is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The Clock Correction Failed Counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

CAUTION

The transition to HALT state is prevented if the FLXAnFRSUCC1.HCSE bit is not set to 1.

18.3.5.2 Passive to Active Counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. FLXAnFRSUCC1.PTA[4:0] defines the number of consecutive valid even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If FLXAnFRSUCC1.PTA[4:0] is set to 0 the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

18.3.5.3 HALT Command

In case the Host wants to stop FlexRay communication of the local node it can bring the CC into HALT state by asserting the HALT command. This can be done by writing 0110 to bits FLXAnFRSUCC1.CMD[3:0]. In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to assure that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from flags FLXAnFRCCSV.PSL[5:0].

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state the POC transits to HALT state at the end of the current cycle. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000” = command_not_accepted and bit FLXAnFREIR.CNA is set to 1. If enabled an interrupt to the Host is generated.

18.3.5.4 FREEZE Command

In case the Host detects a severe error condition it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing 0111 to bits FLXAnFRSUCC1.CMD[3:0]. The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from flags FLXAnFRCCSV.PSL[5:0].

CAUTION

When transfer is stopped by a FREEZE or READY command and then restarted as a leading ColdStart node, the startup frame may not be transmitted in cycle 0. This depends on the internal state of the FlexRay module. Such cases arise when the startup frame is set in a slot 1 to slot 7. This does not occur in a ColdStart after the reset of the microcontroller. Even if the above situation arises, the ColdStart will succeed on the second trial. Repetition prolongs the overall ColdStart process, but the ColdStart will not be obstructed by the earlier situation. To avoid this effect, allocate startup and sync frames to static slot 8 or a slot with a higher number.

18.3.6 Communication Controller States

18.3.6.1 Communication Controller State Diagram

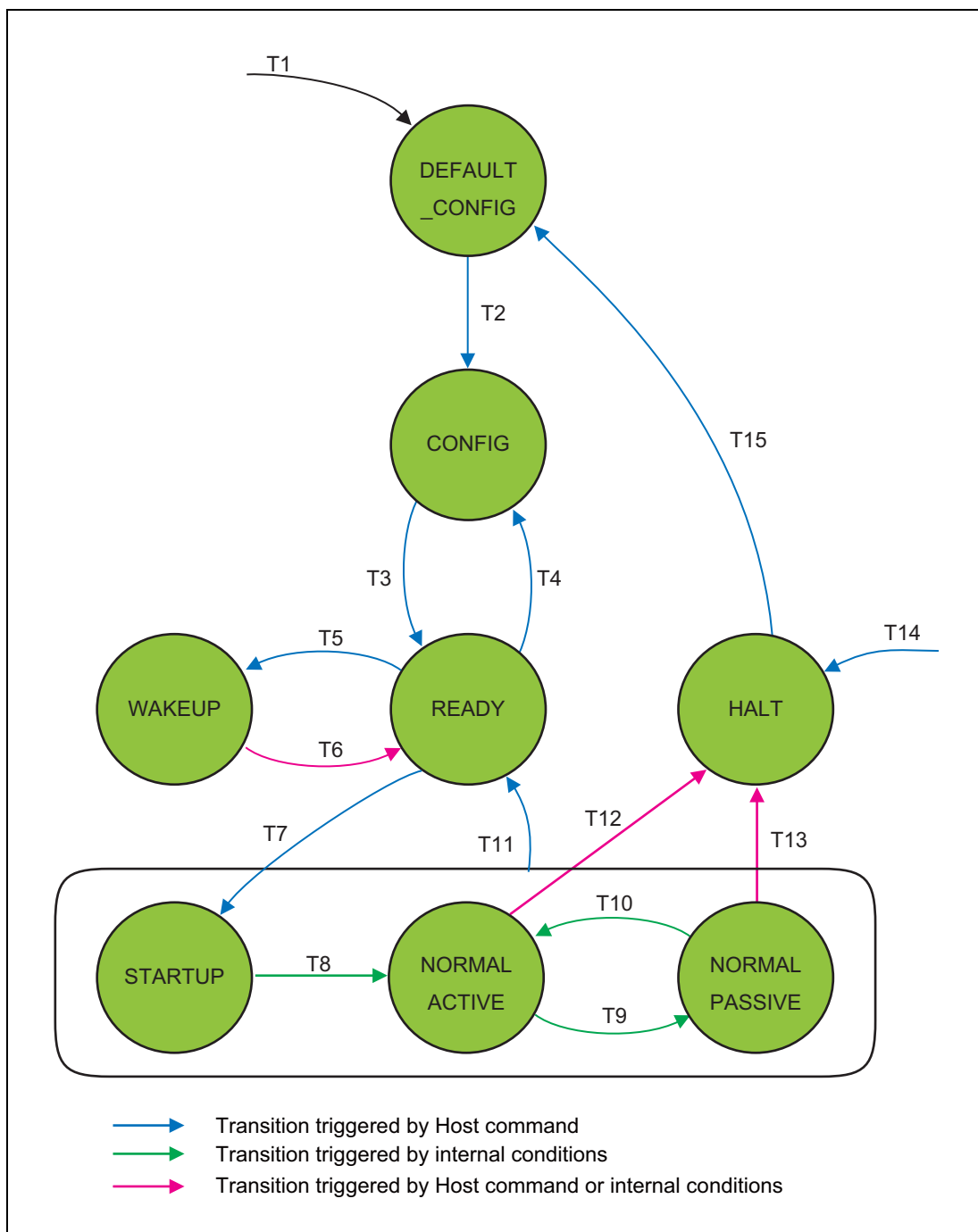


Figure 18.7 Overall State Diagram of FlexRay Communication Controller

State transitions are controlled by reset, rxda_extfxr, rxdb_extfxr, by the POC state machine, and by the CHI Command Vector FLXAnFRSUCC1.CMD[3:0].

The CC transits from all states to HALT state after application of the FREEZE command (FLXAnFRSUCC1.CMD[3:0] = “0111”).

Table 18.96 State Transitions of FlexRay Overall State Machine

| T# | Condition | From | To |
|----|--|--|----------------|
| 1 | Reset | All States | DEFAULT_CONFIG |
| 2 | Command CONFIG: FLXAnFRSUCC1.CMD[3:0] = "0001" | DEFAULT_CONFIG | CONFIG |
| 3 | Unlock sequence followed by command READY: FLXAnFRSUCC1.CMD[3:0] = "0010" | CONFIG | READY |
| 4 | Command CONFIG: FLXAnFRSUCC1.CMD[3:0] = "0001" | READY | CONFIG |
| 5 | Command WAKEUP: FLXAnFRSUCC1.CMD[3:0] = "0011" | READY | WAKEUP |
| 6 | Complete transmission of wakeup pattern, received WUP or received frame header, wakeup collision or command READY: FLXAnFRSUCC1.CMD[3:0] = "0010" | WAKEUP | READY |
| 7 | Command RUN: FLXAnFRSUCC1.CMD[3:0] = "0100" | READY | STARTUP |
| 8 | Successful STARTUP | STARTUP | NORMAL_ACTIVE |
| 9 | Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by FLXAnFRSUCC3.WCP[3:0] | NORMAL_ACTIVE | NORMAL_PASSIVE |
| 10 | Number of valid correction terms reached the Passive to Active limit configured by FLXAnFRSUCC1.PTA[4:0] | NORMAL_PASSIVE | NORMAL_ACTIVE |
| 11 | Command READY: FLXAnFRSUCC1.CMD[3:0] = "0010" | STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE | READY |
| 12 | Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF[3:0] when bit FLXAnFRSUCC1.HCSE set to 1 or command HALT, FLXAnFRSUCC1.CMD[3:0] = "0110" | NORMAL_ACTIVE | HALT |
| 13 | Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF[3:0] when bit FLXAnFRSUCC1.HCSE set to 1, or command HALT: FLXAnFRSUCC1.CMD[3:0] = "0110" | NORMAL_PASSIVE | HALT |
| 14 | Command FREEZE: FLXAnFRSUCC1.CMD[3:0] = "0111" | All States | HALT |
| 15 | Command CONFIG: FLXAnFRSUCC1.CMD[3:0] = "0001" | HALT | DEFAULT_CONFIG |

18.3.6.2 DEFAULT_CONFIG State

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The CC enters this state

- When the reset is applied (the reset of the microcontroller or the software reset of the FlexRay module)
- When exiting from HALT state

To leave DEFAULT_CONFIG state, the Host has to write 0001 to bits FLXAnFRSUCC1.CMD[3:0]. The CC then transits to CONFIG state.

18.3.6.3 CONFIG State

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the Host can analyze status information and configuration. Before leaving CONFIG state, the Host has to assure that the configuration is fault-free.

To leave CONFIG state, the Host has to perform the unlock sequence as described in **Section 18.2.3.1, FLXAnFRLCK — FlexRay Lock Register**. Directly after unlocking the CONFIG state the Host has to write bits FLXAnFRSUCC1.CMD[3:0] to enter the next state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1 to FLXAnFRTXRQ4, and status data stored in the Message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state it is also possible to bring the CC into a power saving mode by halting the module clocks (bus clock and sample clock). To do this the Host has to assure that all Message RAM transfers have finished before turning off the clocks.

18.3.6.4 READY State

After unlocking CONFIG state and writing 0010_B to FLXAnFRSUCC1.CMD[3:0], the CC enters READY state. From this state the CC can transit to WAKEUP state and perform a cluster wakeup or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

- When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing 0010_B to FLXAnFRSUCC1.CMD[3:0] (READY command).

The CC exits from this state

- To CONFIG state by writing 0001_B to FLXAnFRSUCC1.CMD[3:0] (CONFIG command)
- To WAKEUP state by writing 0011_B to FLXAnFRSUCC1.CMD[3:0] (WAKEUP command)
- To STARTUP state by writing 0100_B to FLXAnFRSUCC1.CMD[3:0] (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1 to FLXAnFRTXRQ4, and status data stored in the Message RAM are not affected by the transition of the POC from READY to STARTUP state.

18.3.6.5 WAKEUP State

The description below is intended to help configuring wakeup for the FlexRay IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.1.

The CC enters this state

- When exiting from READY state by writing 0011_B to bits FLXAnFRSUCC1.CMD[3:0] (WAKEUP command).

The CC exits from this state to READY state

- After complete non-aborted transmission of wakeup pattern (WUP)
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing 0010_B to bits FLXAnFRSUCC1.CMD[3:0] (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an external wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and CC to perform the cluster wakeup. The CC provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the CC is in CONFIG state by writing the FLXAnFRSUCC1.WUCS bit. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the CC returns to READY state and signals the change of the wakeup status to the Host by setting flag FLXAnFRSIR.WST. The wakeup status vector can be read from FLXAnFRCCSV.WSV[2:0]. If a valid wakeup pattern was received also either flag FLXAnFRSIR.WUPA or flag FLXAnFRSIR.WUPB is set to 1.

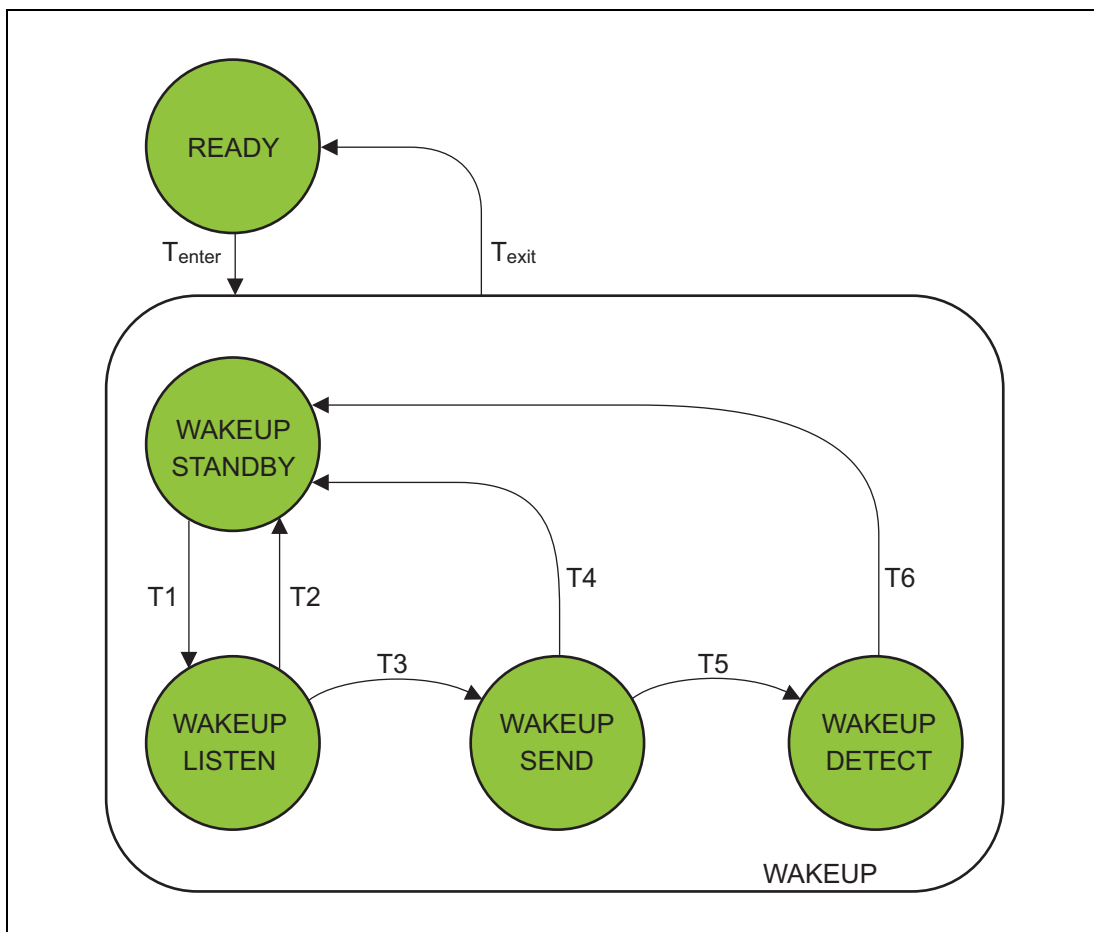


Figure 18.8 Structure of POC State WAKEUP

Table 18.97 State Transitions WAKEUP

| T# | Condition | From | To |
|-------|--|----------------|----------------|
| enter | Host commands change to WAKEUP state by writing 0011 to FLXAnFRSUCC1.CMD[3:0] (WAKEUP command) | READY | WAKEUP |
| 1 | CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state | WAKEUP_STANDBY | WAKEUP_LISTEN |
| 2 | Received WUP on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header on either available channel | WAKEUP_LISTEN | WAKEUP_STANDBY |
| 3 | Timer event | WAKEUP_LISTEN | WAKEUP_SEND |
| 4 | Complete, non-aborted transmission of wakeup pattern | WAKEUP_SEND | WAKEUP_STANDBY |
| 5 | Collision detected | WAKEUP_SEND | WAKEUP_DETECT |
| 6 | Wakeup timer expired or WUP detected on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header received on either available | WAKEUP_DETECT | WAKEUP_STANDBY |
| exit | Wakeup completed (after T2 or T4 or T6) or Host commands change to READY state by writing 0010 to FLXAnFRSUCC1.CMD[3:0] (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state | WAKEUP | READY |

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters Listen Timeout (bits FLXAnFRSUCC2.LT[20:0]) and Listen Timeout Noise (bits FLXAnFRSUCC2.LTN[3:0]). Listen Timeout enables a fast cluster wakeup in case of a noise free environment, while Listen Timeout Noise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by bits FLXAnFRSUCC2.LT[20:0]. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification recommends that two different CCs shall awake the two channels.

(1) Host activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host. The wakeup pattern is detected by the remote BDs and signaled to their local Host.

Wakeup procedure controlled by Host (single-channel wakeup):

- Configure the CC in CONFIG state
 - Select wakeup channel by programming bit FLXAnFRSUCC1.WUCS
- Check local BDs whether a WUP was received
- Activate BD of selected wakeup channel
- Command CC to enter READY state
- Command CC to start wakeup on the configured channel by writing 0011 to bits FLXAnFRSUCC1.CMD[3:0]
 - CC enters WAKEUP
 - CC returns to READY state and signals status of wakeup attempt to the Host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node:
 - In a dual channel cluster wait for WUP on the other channel
 - Set coldstart inhibit flag FLXAnFRCCSV.CSI by writing 1001_B to bits FLXAnFRSUCC1.CMD[3:0] (ALLOW_COLDSTART command)
- Command CC to enter startup by writing 0100_B to bits FLXAnFRSUCC1.CMD[3:0] (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD
- BD triggers power-up of Host (if required)
- BD signals wakeup event to Host
- Host configures its local CC
- If necessary, Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves
- Host commands CC to enter STARTUP state by writing 0100 to bits FLXAnFRSUCC1.CMD[3:0] (RUN command)

(2) Wakeup pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers FLXAnFRPRTC1 and FLXAnFRPRTC2.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)
- Wakeup symbol must be configured identical in all nodes of a cluster
- Wakeup symbol transmit low time configured by bits FLXAnFRPRTC2.TXL[5:0]
- Wakeup symbol idle time used to listen for activity on the bus, configured by bits FLXAnFRPRTC2.TXI[7:0]
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by bits FLXAnFRPRTC1.RWP[5:0] (2 to 63 repetitions)
- Wakeup symbol receive window length configured by bits FLXAnFRPRTC1.RXW[8:0]
- Wakeup symbol receive low time configured by bits FLXAnFRPRTC2.RXL[5:0]
- Wakeup symbol receive idle time configured by bits FLXAnFRPRTC2.RXI[5:0]

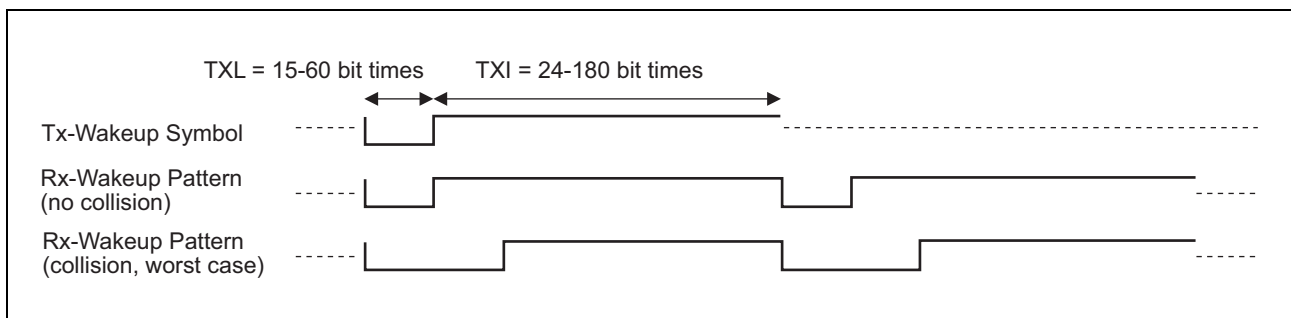


Figure 18.9 Timing of Wakeup Pattern

18.3.6.6 STARTUP State

The description below is intended to help configuring startup for the FlexRay module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the *FlexRay protocol specification v2.1, section 7.2*.

Any node entering STARTUP state that has coldstart capability should assure that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter NORMAL_ACTIVE state via (see **Figure 18.10**):

- Coldstart path initiating the schedule synchronization (LeadingColdstart node)
- Coldstart path joining other coldstart nodes (FollowingColdstart node)
- Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY set to 1. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is sent. In the frame header of the startup frame the startup frame indicator bit is set to 1.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by bits FLXAnFRSUCC1.CSA[4:0].

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

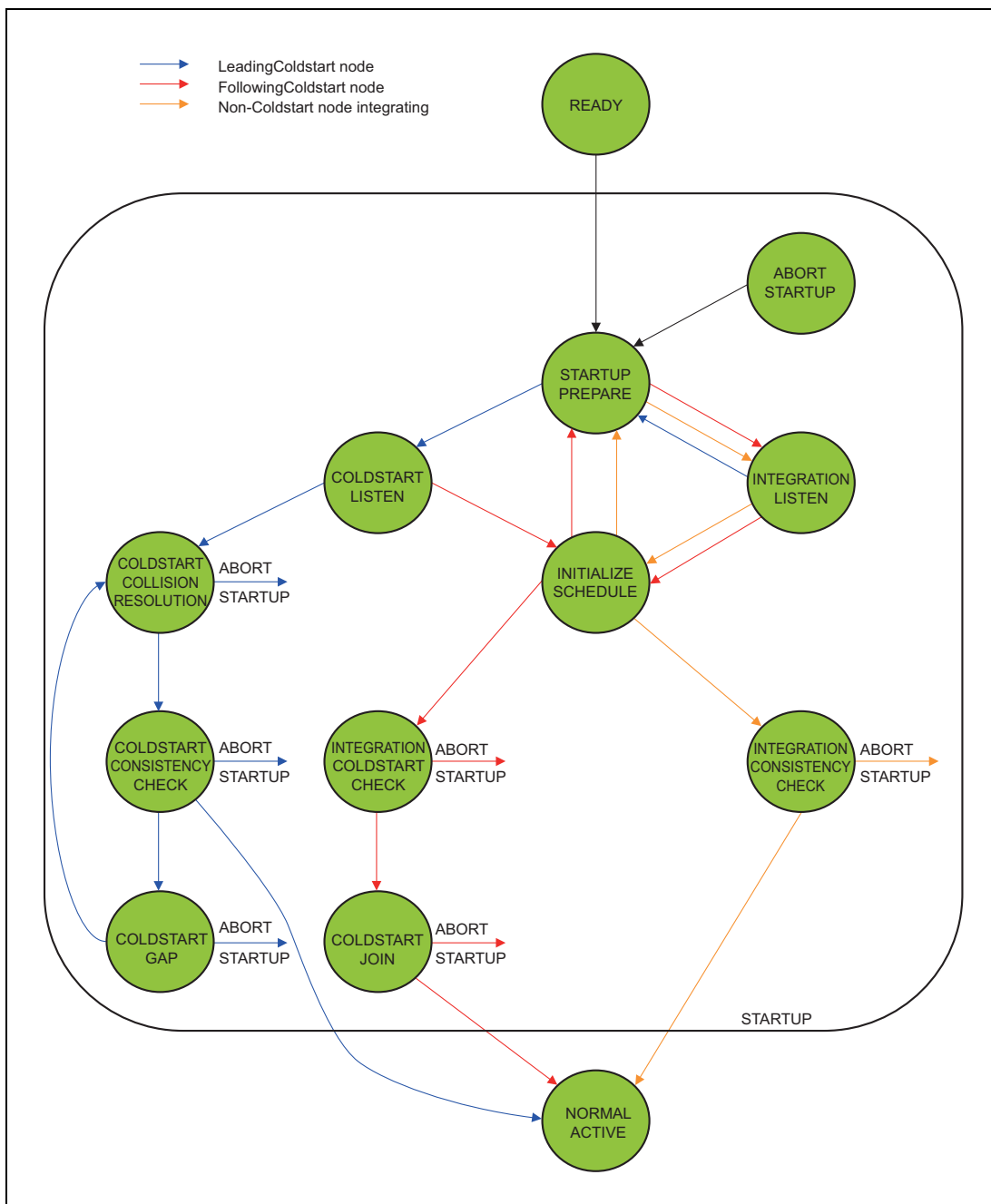


Figure 18.10 State Diagram Time-Triggered Startup

(1) Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit FLXAnFRCCSV.CSI is set to 1, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit FLXAnFRCCSV.CSI is set to 1 whenever the POC enters READY state. The bit has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (bits FLXAnFRSUCC1.CMD[3:0] = “1001_B”)

(2) Startup Timeouts

The CC supplies two different μ T timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are started when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

CAUTION

The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values FLXAnFRSUCC2.LT[20:0] and FLXAnFRSUCC2.LTN[3:0].

(a) Startup Timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming bits FLXAnFRSUCC2.LT[20:0] (see **Section 18.2.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup timeout is:

$$\text{pdListenTimeout} = \text{FLXAnFRSUCC2.LT}[20:0]$$

The startup timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state
- When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

(b) Startup Noise Timeout

At the same time the startup timer is started for the first time (transition from `STARTUP_PREPARE` state to `COLDSTART_LISTEN` state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming bits `FLXAnFRSUCC2.LTN[3:0]` (see **Section 18.2.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup noise timeout:

$$pdListenTimeout \times gListenNoise = FLXAnFRSUCC2.LT[20:0] \times (FLXAnFRSUCC2.LTN[3:0] + 1)$$

The startup noise timer is restarted upon:

- Entering the `COLDSTART_LISTEN` state
- Reception of correctly decoded headers or CAS symbols while the node is in `COLDSTART_LISTEN` state

The startup noise timer is stopped when the `COLDSTART_LISTEN` state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won't be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

(3) Path of Leading Coldstart Node (initiating coldstart)

When a coldstart node enters `COLDSTART_LISTEN`, it listens to its attached channels.

If no communication is detected, the node enters the `COLDSTART_COLLISION_RESOLUTION` state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the `COLDSTART_LISTEN` state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in `COLDSTART_COLLISION_RESOLUTION` state, the node that initiated the coldstart enters the `COLDSTART_CONSISTENCY_CHECK` state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves `COLDSTART_CONSISTENCY_CHECK` and enters `NORMAL_ACTIVE` state.

The number of coldstart attempts that a node is allowed to perform is configured by `FLXAnFRSUCC1.CSA[4:0]`. The number of remaining coldstarts attempts can be read from `FLXAnFRCCSV.RCA[4:0]`. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the `COLDSTART_LISTEN` state only if this value is larger than one and it may enter the `COLDSTART_COLLISION_RESOLUTION` state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

(4) Path of Following Coldstart Node (responding to Leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still available. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state following coldstart nodes begin to transmit their own startup frames and continue to do so in subsequent cycles. Thereby, the leading coldstart node and the nodes joining it can check if their schedules agree with each other. If the clock correction signals any error, the node aborts the integration attempt. If a node in this state sees at least one valid startup frame during all even cycles in this state and at least one valid startup frame pair during all double cycles in this state, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

(5) Path of Non-coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. Clock correction is activated, and if any errors are signaled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

18.3.6.7 NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even / odd cycle pairs required).

In NORMAL_ACTIVE state the CC supports regular communication functions

- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The CC exits from that state to

- HALT state by writing 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command, at the end of the current cycle)
- HALT state by writing 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing 0010_B to bits FLXAnFRSUCC1.CMD[3:0] (READY command)

18.3.6.8 NORMAL_PASSIVE State

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state

- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The CC exits from this state to

- HALT state by writing 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command, at the end of the current cycle)
- HALT state by writing 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE. The transition takes place when bits FLXAnFRCCEV.PTAC[4:0] equals bits FLXAnFRSUCC1.PTA[4:0] - 1
- To READY state by writing 0010_B to bits FLXAnFRSUCC1.CMD[3:0] (READY command)

18.3.6.9 HALT State

In this state all communication (reception and transmission) is stopped.

The CC enters this state

- By writing 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the clock correction failed counter reached the “maximum without clock correction fatal” limit and FLXAnFRSUCC1.HCSE is set to 1
- When exiting from NORMAL_PASSIVE state because the clock correction failed counter reached the “maximum without clock correction fatal” limit and FLXAnFRSUCC1.HCSE is set to 1

The CC exits from this state to DEFAULT_CONFIG state

- By writing 0001_B to bits FLXAnFRSUCC1.CMD[3:0] (CONFIG command)

When the CC enters HALT state, all configuration and status data is maintained for analyzing purposes.

When the Host writes 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command), the CC sets flag FLXAnFRCCSV.HRQ to 1 and enters HALT state at the next end of cycle.

When the Host writes 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command), the CC enters HALT state immediately and sets flag FLXAnFRCCSV.FSI to 1.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL[5:0].

18.3.7 Network Management

The accrued Network Management (NM) vector can be read from registers FLXAnFRNMV1 to FLXAnFRNMV3. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (PPI) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by bits FLXAnFRNEMC.NML[3:0]. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set as 1, bit PPIT in the header section of the respective transmit buffer has to be set to 1 via bit FLXAnFRWRHS1.PPIT. In addition the Host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the Host.

CAUTIONS

1. **In case a message buffer is configured for transmission / reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by the FLXAnFRNEMC.NML[3:0] bits.**
 2. **When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case FLXAnFRNMV1 to FLXAnFRNMV3 holds the value from the cycle before.**
-

18.3.8 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated / transmitted if the required matches occur.

Filtering is done on:

- Slot Counter
- Cycle Counter
- Channel ID

The following filter combinations for acceptance / transmit filtering are allowed:

- Slot Counter + Channel ID
- Slot Counter + Cycle Counter + Channel ID

All configured filters must match in order to store a received message in a message buffer.

CAUTION

For the FIFO the acceptance filter is configured by the FIFO Rejection Filter (FLXAnFRFRF) and the FIFO Rejection Filter Mask (FLXAnFRFRFM).

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

18.3.8.1 Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID and channel ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

18.3.8.2 Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 resp. 1 is configured to hold the startup / sync frame or the single slot frame by bits FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM, cycle counter filtering for message buffer 0 resp. 1 shall be disabled.

CAUTION

Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is not allowed.

The set of cycle numbers belonging to a cycle set is determined as described in **Table 18.98**.

Table 18.98 Definition of cycle set

| Cycle Code | Matching Cycle Counter Values |
|------------|---|
| 0b000000x | all Cycles |
| 0b000001c | every second Cycle at (Cycle Count) mod2 = c |
| 0b00001cc | every fourth Cycle at (Cycle Count) mod4 = cc |
| 0b0001ccc | every eighth Cycle at (Cycle Count) mod8 = ccc |
| 0b001cccc | every sixteenth Cycle at (Cycle Count) mod16 = cccc |
| 0b01ccccc | every thirty-second Cycle at (Cycle Count) mod32 = ccccc |
| 0b1cccccc | every sixty-fourth Cycle at (Cycle Count) mod64 = ccccccc |

Table 18.99 below gives some examples for valid cycle sets to be used for cycle counter filtering.

Table 18.99 Examples for valid cycle sets

| Cycle Code | Matching Cycle Counter Values |
|------------|-------------------------------|
| 0b0000011 | 1-3-5-7-.... -63 |
| 0b0000100 | 0-4-8-12-.... -60 |
| 0b0001110 | 6-14-22-30-.... -62 |
| 0b0011000 | 8-24-40-56 |
| 0b0100011 | 3-35 |
| 0b1001001 | 9 |

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Channel ID and frame ID must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Channel ID and frame ID must also be met.

18.3.8.3 Channel ID Filtering

There is a 2-bit channel filtering field (CH) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see **Table 18.100**).

Table 18.100 Channel filtering configuration

| CH[1:0] | Transmit Buffer transmit frame | Receive Buffer store valid receive frame |
|---------|--|--|
| 00 | no transmission | ignore frame |
| 01 | on channel A | received on channel A |
| 10 | on channel B | received on channel B |
| 11 | on both channels (static segment only) | received on channel A or B (store first semantically valid frame, static segment only) |

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (CH = “11_B”).

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (CH = “11”).

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted resp. received frames are ignored (same function as CH = “00”).

18.3.8.4 FIFO Filtering

For FIFO filtering registers FLXAnFRFRF and FLXAnFRFRFM are used. The FIFO filter consists of channel filter bits FLXAnFRFRF.CH[1:0], frame ID filter bits FLXAnFRFRF.FID[10:0], and cycle counter filter bits FLXAnFRFRF.CYF[6:0]. Registers FLXAnFRFRF and FLXAnFRFRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

18.3.9 Transmit Process

18.3.9.1 Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

18.3.9.2 Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

The start of latest transmit configured by bits `FLXAnFRMHDC.SLT[12.0]` defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

18.3.9.3 Transmit Buffers

FlexRay message buffers can be configured as transmit buffers by programming bit `CFG` in the header section of the respective message buffer to 1 via `FLXAnFRWRHS1`.

There exist the following possibilities to assign a transmit buffer to the CC channels:

- Static segment: channel A or channel B,
channel A and channel B
- Dynamic segment: channel A or channel B

Message buffer 0 resp. 1 is dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by `FLXAnFRSUCC1.TXST`, `FLXAnFRSUCC1.TXSY`, and `FLXAnFRSUCC1.TSM`. In this case, it can be reconfigured in `DEFAULT_CONFIG` or `CONFIG` state only. This ensures that any node transmits at most one startup / sync frame per communication cycle. Transmission of startup / sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of bits `FLXAnFRMRC.SEC[1.0]` (see **Section 18.3.12.1, Reconfiguration of Message Buffers**). Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The CC does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the Host has to set the `PPIT` bit in the header section of the respective message buffer to 1 and write the network management information to the data section of the message buffer (see **Section 18.3.7, Network Management**).

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by

FLXAnFRMHDC.SFDL[6.0], the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is “0000_H”.

CAUTION

In case of an odd payload length (PLC = 1, 3, 5, so on) the application has to write zero to the last 16 bit of the message buffers data section to ensure that the padding pattern is 0000_H.

Each transmit buffer provides a transmission mode flag TXM that allows the Host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In single-shot mode the CC resets the respective TXR flag to 0 after transmission has completed. Now the Host may update the transmit buffer.

In continuous mode, the CC does not reset the respective transmission request flag TXR to 0 after successful transmission. In this case a frame is sent out each time the filter criteria match. The TXR flag can be reset to 0 by the Host by writing the respective message buffer number to the FLXAnFRIBCR register while bit FLXAnFRIBCM.STXRH is set to 0.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

18.3.9.4 Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the Message RAM via registers FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write the data section of the transmit buffer via register FLXAnFRWRDSm
- Transfer the configuration and message data from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR
- If configured in register FLXAnFRIBCM, the transmission request flag TXR for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective TXR bit (TXR = 0) in the FLXAnFRTRXQ1 to FLXAnFRTRXQ4 registers (single-shot mode only).

After transmission has completed, the respective TXR flag in the FLXAnFRTXRQ1 to FLXAnFRTXRQ4 registers is reset to 0 (single-shot mode), and, if bit MBI in the header section of the message buffer is set to 1, bit FLXAnFRSIR.TXI is set to 1. If enabled, an interrupt is generated.

18.3.9.5 Null Frame Transmission

If in static segment the Host does not set the transmission request flag to 1 before transmit time, the CC transmits a null frame with the null frame indication bit set to 0 and the payload data set to zero.

In the following cases the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (TXR = 0) to 1.
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status FLXAnFRMBS is updated.

Null frames are not transmitted in the dynamic segment.

18.3.10 Receive Process

18.3.10.1 Dedicated Receive Buffers

A portion of the FlexRay message buffers can be configured as dedicated receive buffers by programming bit CFG in the header section of the respective message buffer to 0 via FLXAnFRWRHS1.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: channel A or channel B,
channel A and channel B (the CC stores the first semantically valid frame)
- Dynamic segment: channel A or channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of bits FLXAnFRMRC.SEC[1.0] (see **Section 18.3.12.1, Reconfiguration of Message Buffers**). If a message buffer is reconfigured (header section updated) during runtime it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

18.3.10.2 Frame Reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the Message RAM via registers FLXAnFRWRHS1 to FLXAnFRWRHS3
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective ND flag in the FLXAnFRNDAT1 to FLXAnFRNDAT4 registers is set to 1, and, if bit MBI in the header section of that message buffer is set to 1, bit FLXAnFRSIR.RXI is set to 1. If enabled, an interrupt is generated.

In case that bit ND was already set to 1 when the Message Handler updates the message buffer, bit FLXAnFRMBS.MLST of the respective message buffer is set to 1 and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status FLXAnFRMBS is updated.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1 to FLXAnFRMBSC4 registers is set to 1, and if bit MBI in the header section of that message buffer is set, bit FLXAnFRSIR.MBSI is set to 1. If enabled an interrupt is generated.

If the payload length of a received frame PLR is longer than the value programmed by PLC in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the Message RAM via the Output Buffer, proceed as described in **Section 18.3.12.2 (2), Data Transfer from Message RAM to Output Buffer.**

CAUTION

The ND and MBC flags are automatically cleared by the Message Handler when the payload data and the header of a received message have been transferred to the Output Buffer, respectively.

18.3.10.3 Null Frame Reception

The payload segment of a received null frame is not copied into the matching dedicated receive buffer. If a null frame has been received, only the message buffer status FLXAnFRMBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1 to FLXAnFRMBSC4 register is set to 1, and if bit MBI in the header section of that message buffer is set to 1, flag FLXAnFRSIR.MBSI is set to 1. If enabled, an interrupt is generated.

18.3.11 FIFO Function

18.3.11.1 Description

A portion of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FLXAnFRMRC.FFB[7.0] and ending with the message buffer referenced by FLXAnFRMRC.LCB[7.0]. Up to 127 message buffers can be assigned to the FIFO.

Every valid incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the message buffer status FLXAnFRMBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. When Flag FLXAnFRSIR.RFNE is set to 1, it shows that the FIFO is not empty. When bit FLXAnFRSIR.RFCL is set to 1, it shows that the receive FIFO fill level (FLXAnFRFSR.RFFL[7.0]) is equal or greater than the critical level as configured by FLXAnFRFCL.CL. When bit FLXAnFREIR.RFO is set to 1, it shows that a FIFO overrun has been detected. If interrupts are enabled, interrupt requests are generated.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag FLXAnFREIR.RFO to 1.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag FLXAnFRSIR.RFNE is set to 1. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in **Figure 18.11** for a three message buffer FIFO.

The programmable FlexRay FIFO Rejection Filter (FLXAnFRFRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit FLXAnFRFRF.RSS is set to 1, all messages received in the static segment are rejected by the FIFO. If bit FLXAnFRFRF.RNF is set to 1, received null frames are not stored in the FIFO.

The FlexRay FIFO Rejection Filter Mask (FLXAnFRFRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked 'don't care' for rejection filtering.

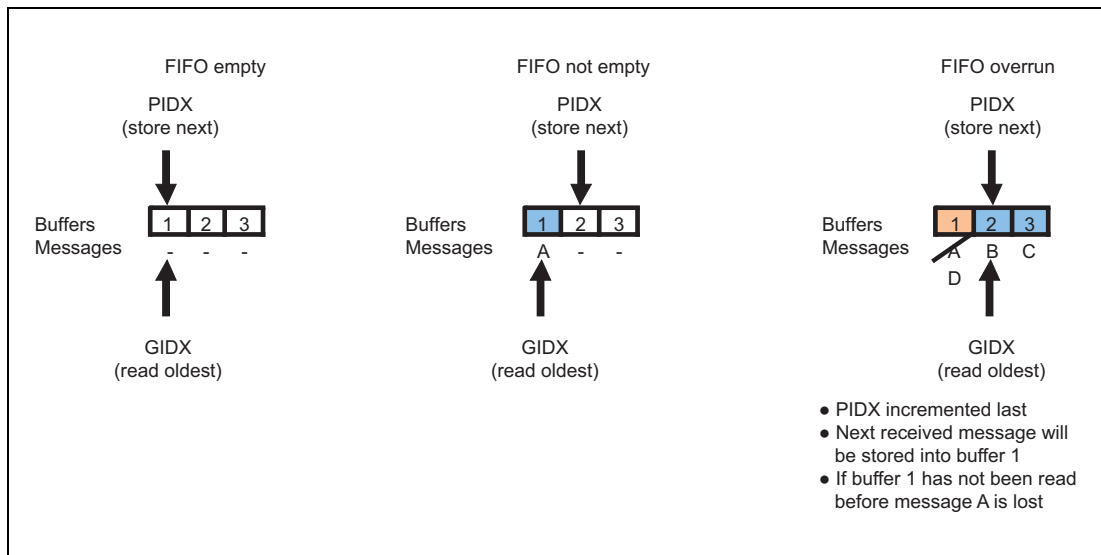


Figure 18.11 FIFO Status: Empty, Not Empty, Overrun

18.3.11.2 Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured should be programmed to the same value via bits FLXAnFRWRHS2.PLC[6.0]. The data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via bits FLXAnFRWRHS3.DP[10.0].

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of DP and PLC, irrelevant.

CAUTIONS

1. It is recommended to program the MBI bits of the message buffers belonging to the FIFO to 0 via FLXAnFRWRHS1.MBI to avoid generation of RX interrupts.
2. If the payload length of a received frame is longer than the value programmed by FLXAnFRWRHS2.PLC in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.

18.3.11.3 Access to the FIFO

(1) When the output buffer is used:

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by FLXAnFRMRC.FFB[7.0]) to the register FLXAnFROBCR. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

(2) When the data transfer function is used:

The message received in FIFO can be transferred to the Local RAM/Global RAM by using the output data transfer function. For the output data transfer function, see **Section 18.3.16.2, Output Data Transfer**, Output Data Transfer.

18.3.12 Message Handling

The Message Handler controls data transfers between the Input / Output Buffer and the Message RAM and between the Message RAM and the two TBFs.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to bits FLXAnFRGTUC7.NSS[9:0]. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from bits FLXAnFRGTUC7.NSS[9:0] + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

Access of the Host to the message buffer contents using the input or output buffer function is described in this subsection. Access to the message buffer contents using the data transfer function is mentioned in **Section 18.3.16, Usage of Data Transfer**.

18.3.12.1 Reconfiguration of Message Buffers

In case that an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via Input Buffer registers FLXAnFRWRHS1 to FLXAnFRWRHS3.

Reconfiguration has to be enabled via control bit FLXAnFRMRC.SEC in the Message RAM Configuration register.

If a message buffer has not been transmitted / updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission / reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted / updated from a received frame in the cycle where it was reconfigured.

The Message RAM is scanned according to **Table 18.101** below.

Table 18.101 Scan of Message RAM

| Start of Scan in Slot | Scan for Slots |
|-----------------------|--------------------------|
| 1 | 2 to 15, 1 (next cycle) |
| 8 | 16 to 23, 1 (next cycle) |
| 16 | 24 to 31, 1 (next cycle) |
| 24 | 32 to 39, 1 (next cycle) |
| | |

A Message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the Message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the Message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the Message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by bits FLXAnFRMRC.FDB[7:0]. In case a Message RAM scan starts while the CC is in dynamic segment, the scan starts with the message buffer number configured by bit FLXAnFRMRC.FDB.

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the following has to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the “Static Buffers”, it will only be found if it is reconfigured before the last Message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the “Static + Dynamic Buffers”, it will be found if it is reconfigured before the last Message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the Message RAM scan. In case the Message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

CAUTION

Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted / updated from a received frame.

18.3.12.2 Host access to Message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target / source message buffer to be accessed to FLXAnFRIBCR or FLXAnFROBCR register.

The FLXAnFRIBCM and FLXAnFROBCM registers can be used to write / read header and data section of the selected message buffer separately.

If bit FLXAnFRIBCM.STXR is set to 1, the transmission request flag TXR of the selected message buffer is automatically set to 1 after the message buffer has been updated. If bit FLXAnFRIBCM.STXR is reset to 0, the transmission request flag TXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and Output Buffer (OBF) are build up as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host / OBF Host), while the other half (IBF Shadow / OBF Shadow) is accessed by the Message Handler for data transfers between IBF / OBF and Message RAM.

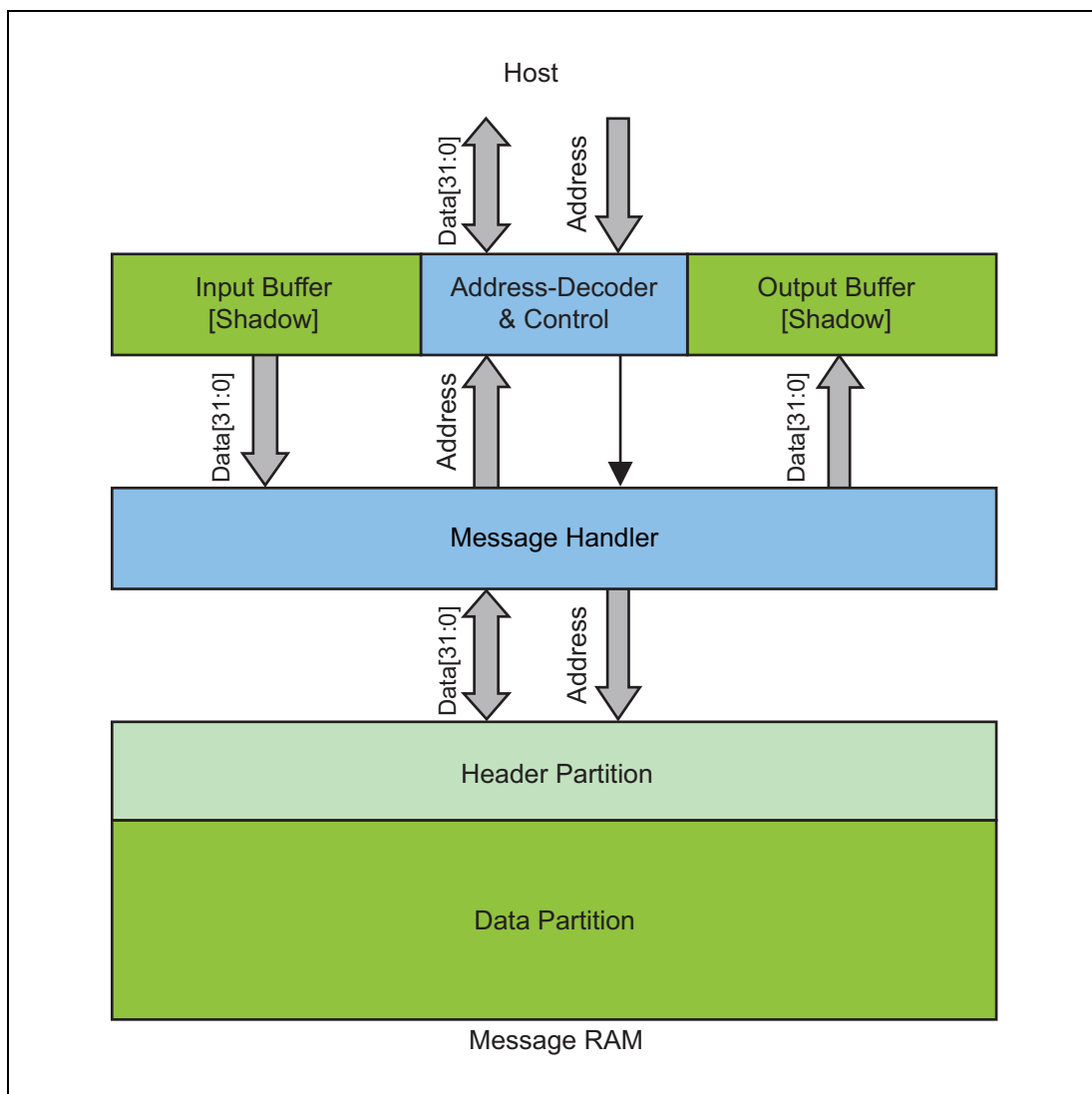


Figure 18.12 Host access to Message RAM

(1) Data Transfer from Input Buffer to Message RAM

To configure / update a message buffer in the Message RAM, the Host has to write the data to FLXAnFRWRDSm and the header to FLXAnFRWRHS1 to FLXAnFRWRHS3. The specific action is selected by configuring the FlexRay Input Buffer Command Mask FLXAnFRIBCM.

When the Host writes the number of the target message buffer in the Message RAM to bits FLXAnFRIBCR.IBRH[6:0], IBF Host and IBF Shadow are swapped (see **Figure 18.13**).

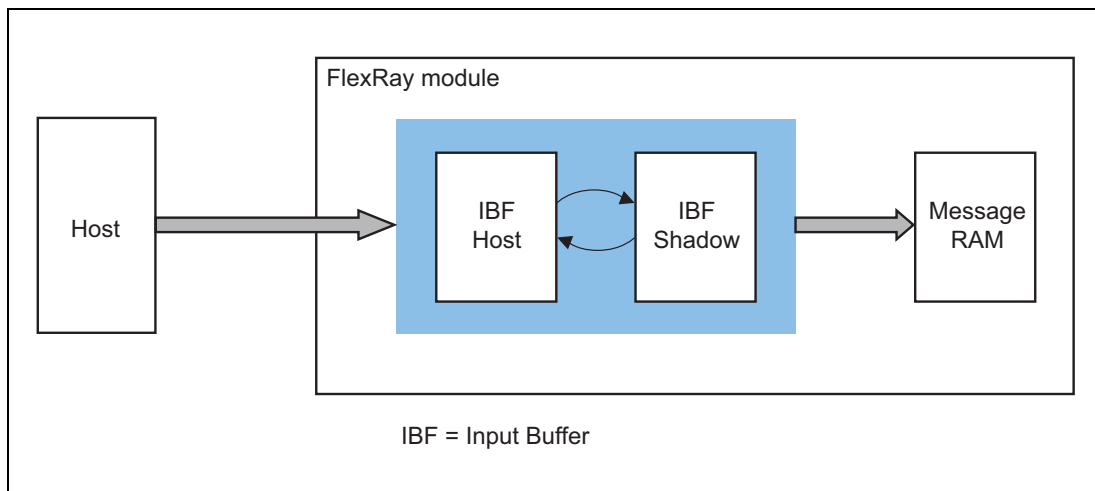


Figure 18.13 Double Buffer Structure Input Buffer

In addition the bits in the FLXAnFRIBCM and FLXAnFRIBCR registers are also swapped to keep them attached to the respective IBF section (see **Figure 18.14**).

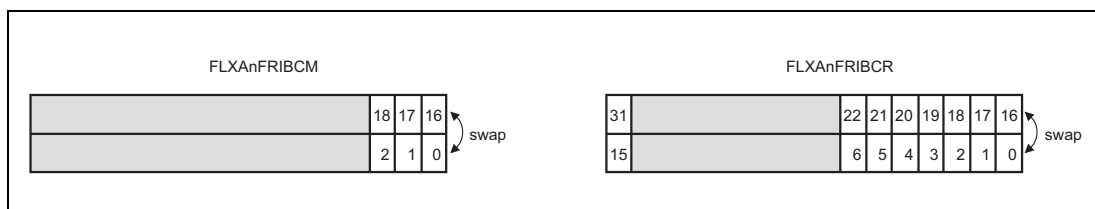


Figure 18.14 Swapping of FLXAnFRIBCM and FLXAnFRIBCR bits

With this write operation bit FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by bits FLXAnFRIBCR.IBRS[6:0].

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to bits FLXAnFRIBCR.IBRH[6:0].

If a write access to bits FLXAnFRIBCR.IBRH[6:0] occurs while FLXAnFRIBCR.IBSYS is 1, FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, FLXAnFRIBCR.IBSYH is reset to 0, FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under bits FLXAnFRIBCR.IBRH[6:0] and FLXAnFRIBCR.IBRS[6:0] and the command mask flags are also swapped.

Example of a 8/16/32-bit Host access sequence:

Configure / update k-th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSm
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH[6:0]

Configure / update (k+1)th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSm
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: set bits FLXAnFRIBCR.IBRH[6:0]

CAUTION

Any write access to IBF while bit FLXAnFRIBCR.IBSYH is 1 will set bit FLXAnFREIR.IIBA to 1. In this case the write access has no effect.

Table 18.102 Assignment of FLXAnFRIBCM Bits

| Pos. | Access | Bit | Function |
|------|--------|-------|---|
| 18 | R | STXRS | Set Transmission Request Shadow ongoing or finished |
| 17 | R | LDSS | Load Data Section Shadow ongoing or finished |
| 16 | R | LHSS | Load Header Section Shadow ongoing or finished |
| 2 | R/W | STXRH | Set Transmission Request Host |
| 1 | R/W | LDSH | Load Data Section Host |
| 0 | R/W | LHSH | Load Header Section Host |

Table 18.103 Assignment of FLXAnFRIBCR Bits

| Pos. | Access | Bit | Function |
|----------|--------|-----------|--|
| 31 | R | IBSYS | IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM |
| 22 to 16 | R | IBRS[6:0] | IBF Request Shadow, number of message buffer currently / lately updated |
| 15 | R | IBSYH | IBF Busy Host, transfer request pending for message buffer referenced by IBRH[6:0] |
| 6 to 0 | R/W | IBRH[6:0] | IBF Request Host, number of message buffer to be updated next |

(2) Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to register FLXAnFROBCR to trigger the data transfer as configured in FLXAnFROBCM. After the transfer has completed, the Host can read the transferred data from FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS.

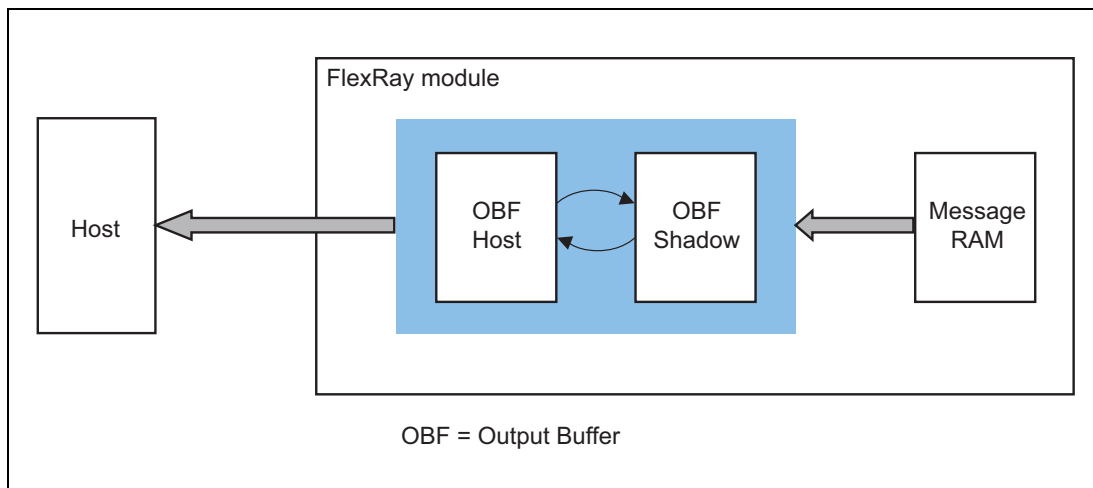


Figure 18.15 Double buffer structure Output Buffer

OBF Host and OBF Shadow as well as bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS, FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH and bits FLXAnFROBCR.OBRS[6:0], FLXAnFROBCR.OBRH[6:0] are swapped under control of bits FLXAnFROBCR.VIEW and FLXAnFROBCR.REQ.

Writing bit FLXAnFROBCR.REQ to 1 copies bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS and bits FLXAnFROBCR.OBRS[6:0] to an internal storage (see **Figure 18.16**).

After setting bit FLXAnFROBCR.REQ to 1, FLXAnFROBCR.OBSYS is set to 1, and the transfer of the message buffer selected by bits FLXAnFROBCR.OBRS[6:0] from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, bit FLXAnFROBCR.OBSYS is set back to 0. Bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW can only be set to 1 while bit FLXAnFROBCR.OBSYS is 0.

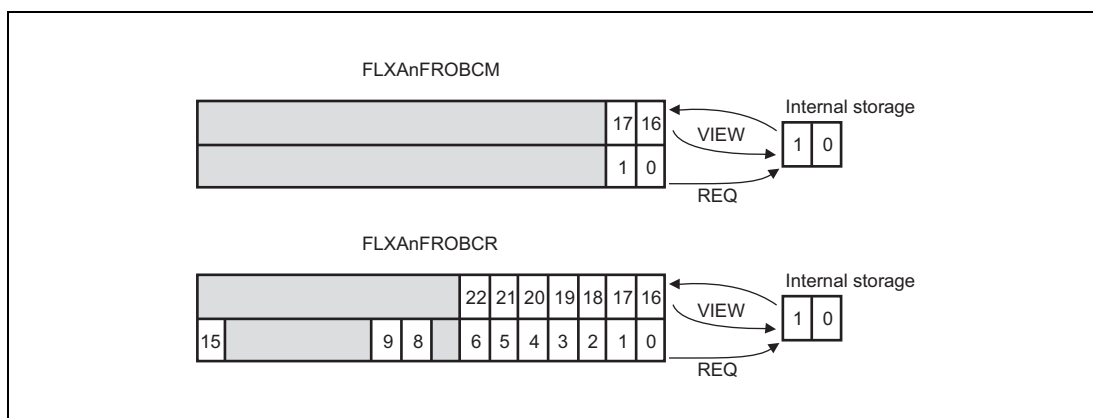


Figure 18.16 Swapping of FLXAnFROBCM and FLXAnFROBCR bits

OBF Host and OBF Shadow are switched by setting bit FLXAnFROBCR.VIEW to 1 while bit FLXAnFROBCR.OBSYS is 0 (see **Figure 18.15**).

In addition, bits FLXAnFROBCR.OBRH[6:0] and bits FLXAnFROBCM.RHSH and FLXAnFROBCM.RDSH are switched with the registers internal storage thus assuring that the message buffer number stored in bits FLXAnFROBCR.OBRH[6:0] and the mask configuration stored in FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH match the transferred data stored in OBF Host (see **Figure 18.16**).

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

If bits REQ and VIEW are set to 1 with the same write access while FLXAnFROBSYS is 0, FLXAnFROBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards bits FLXAnFROBCR.OBRS[6:0] are copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Example of an 8/16/32-bit Host access to a single message buffer:

If a single message buffer has to be read out, two separate write accesses to FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are necessary:

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS
- Request transfer of message buffer to OBF Shadow by writing FLXAnFROBCR.OBRS[6:0] and FLXAnFROBCR.REQ (in case of an 8-bit Host interface, FLXAnFROBCR.OBRS[6:0] have to be written before FLXAnFROBCR.REQ).
- Wait until FLXAnFROBCR.OBSYS is reset
- Toggle OBF Shadow and OBF Host by writing FLXAnFROBCR.VIEW = 1
- Read out transferred message buffer by reading FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Example of an 8/16/32-bit Host access sequence:

Request transfer of 1st message buffer to OBF Shadow

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing bits FLXAnFROBCR.OBRS[6:0] and FLXAnFROBCR.REQ (in case of accessing to the FLXAnFROBCR register in 8-bit units, bits FLXAnFROBCR.OBRS[6:0] have to be written before FLXAnFROBCR.REQ).

Toggle OBF Shadow and OBF Host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 2nd message buffer
- Toggle OBF Shadow and OBF Host and start transfer of 2nd message buffer to OBF Shadow simultaneously by writing FLXAnFROBCR.OBRS[6:0] of 2nd message buffer, bits FLXAnFROBCR.REQ, and FLXAnFROBCR.VIEW (in case of and 8-bit access to register FLXAnFROBCR, bits FLXAnFROBCR.OBRS[6:0] have to be written before bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW).
- Read out 1st transferred message buffer by reading FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Demand access to last requested message buffer without request of another message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0
- Demand access to last transferred message buffer by writing FLXAnFROBCR.VIEW
- Read out last transferred message buffer by reading FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Table 18.104 Assignment of FLXAnFROBCM bits

| Pos. | Access | Bit | Function |
|------|--------|------|--|
| 17 | R | RDSH | Data Section available for Host access |
| 16 | R | RHSH | Header Section available for Host access |
| 1 | R/W | RDSS | Read Data Section Shadow |
| 0 | R/W | RHSS | Read Header Section Shadow |

Table 18.105 Assignment of FLXAnFROBCR bits

| Pos. | Access | Bit | Function |
|----------|--------|-----------|--|
| 22 to 16 | R | OBRH[6:0] | OBF Request Host, number of message buffer available for Host access |
| 15 | R | OBSYS | OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow |
| 9 | R/W | REQ | Request Transfer from Message RAM to OBF Shadow |
| 8 | R/W | VIEW | View OBF Shadow, swap OBF Shadow and OBF Host |
| 6 to 0 | R/W | OBRS[6:0] | OBF Request Shadow, number of message buffer for next request |

18.3.12.3 FlexRay Protocol Controller Access to Message RAM

The two TBFs (A,B) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each TBF is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If e.g. the Message Handler writes the next message to be sent to TBF Tx, the FlexRay Channel Protocol Controller can access TBF Rx to store the message it is actually receiving. During transmission of the message stored in TBF Tx, the Message Handler transfers the last received message stored in TBF Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the TBFs and the shift registers of the FlexRay Channel Protocol Controllers are done in words of 32 bit. This enables the use of a 32 bit shift register independent of the length of the FlexRay messages.

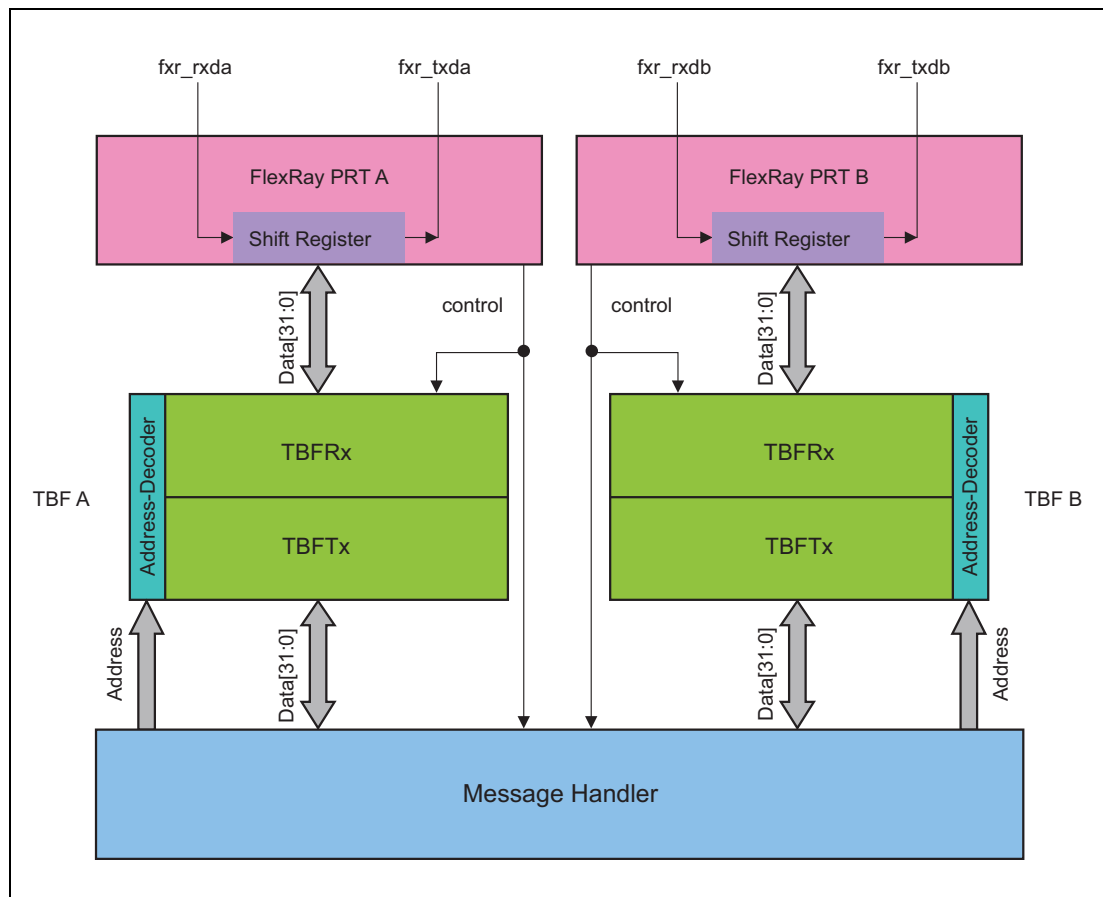


Figure 18.17 Access to TBFs

18.3.13 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception / transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is able to store up to 2048 32-bit words. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0 to 254), the Message RAM has a structure as shown in **Figure 18.18**.

When a message buffer of the data section to be allocated immediately after to the header partition is set as a reception buffer (bit FLXAnFRWRHS1.CFG to 0) or reception FIFO buffer, set a 32-bit unused area (minimum) at the beginning of the data section. In this case, the data partition is allowed to start at Message RAM word number: $((\text{bits FLXAnFRMRC.LCB}[7:0] + 1) \times 4) + 1$.

When a message buffer of the data section to be allocated immediately after to the header partition is set as a transmit buffer (bit FLXAnFRWRHS1.CFG to 1), the data partition is allowed to start at Message RAM word number: $(\text{bits FLXAnFRMRC.LCB}[7:0] + 1) \times 4$.

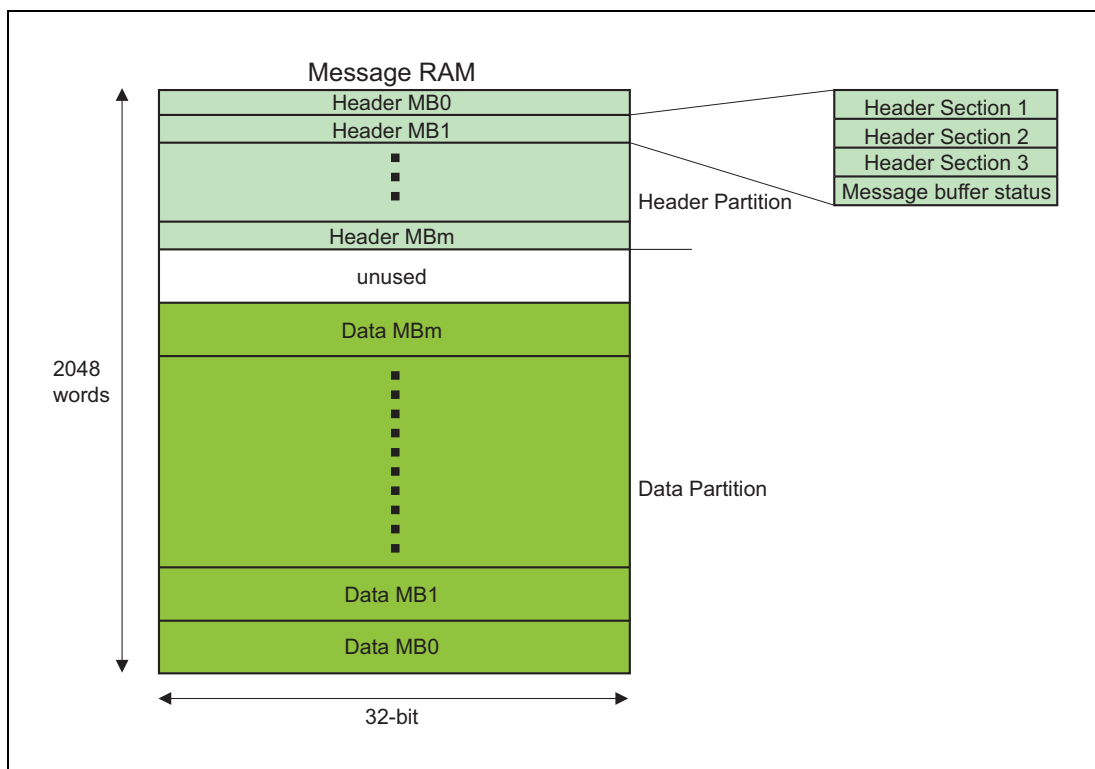


Figure 18.18 Configuration Example of Message Buffers in the Message RAM

Header Partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32 bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

CAUTION

header partition + data partition may not occupy more than 2048 32-bit words.

18.3.13.1 Header Partition

The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in **Table 18.106** below. Configuration of the header sections of the message buffers is done via IBF (FLXAnFRWRHS1 to FLXAnFRWRHS3). Read access to the header sections is done via OBF (FLXAnFRRDHS1 to FLXAnFRRDHS3 + FLXAnFRMBS). The data pointer has to be calculated by the user to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 32-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host.

Payload Length Received PLR, Receive Cycle Count RCC, Received on Channel Indicator RCI, Startup Frame Indicator SFI, Sync Frame Indicator SYN, Null Frame Indicator NFI, Payload Preamble Indicator PPI, and Reserved Bit RES are updated from received valid data frames only.

Table 18.106 Header Section of a Message Buffer in the Message RAM

| Bit Word | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|----------|-----|-------------------------|-------------|-------------|------------------|-------------|-------------|---------------------------|----|---------------------|----|----|----|----|----|-------------|-------------|------------------|--|--------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|---|---|--|
| 0 | | | M B I | T X M | P P I T | C F G | | CH | | Cycle Code | | | | | | | | | | | | | | | | Frame ID | | | | | | | |
| 1 | | Payload Length Received | | | | | | Payload length Configured | | | | | | | | | | | Tx Buffer: Header CRC Configured Rx Buffer: Header CRC Received | | | | | | | | | | | | | | |
| 2 | | | R E S | P P I | N F I | S Y N | S F I | S R C I | | Receive Cycle count | | | | | | | | | | Data Pointer | | | | | | | | | | | | | |
| 3 | | | R E S | P P I | N F I | S Y N | S F I | S R C I | | Cycle Count Status | | | | | | F T B | F T A | M L S T | E S B | E S A | T C I B | T C I A | S V O B | S V O A | C E O B | C E O A | S E O B | S E O A | V F R B | V F R A | | | |
| ... | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | |
|--|----------------------------------|
| | Frame Configuration |
| | Filter Configuration |
| | Message Buffer Contrd |
| | Message RAM Configuration |
| | Updated from received Data Frame |
| | Message Buffer Status (MBS) |
| | unused |

(1) Header section 1 (word 0)

Write access via FLXAnFRWRHS1, read access via FLXAnFRRDHS1:

- Frame ID
 - Slot counter filtering configuration
- Cycle Code
 - Cycle counter filtering configuration
- CH
 - Channel filtering configuration
- CFG
 - Message buffer direction configuration: receive / transmit
- PPIT
 - Payload Preamble Indicator Transmit
- TXM
 - Transmit mode configuration: single-shot / continuous
- MBI
 - Message buffer receive / transmit interrupt enable

(2) Header section 2 (word 1)

Write access via FLXAnFRWRHS2, read access via FLXAnFRRDHS2:

- Header CRC
 - Transmit Buffer: Configured by the Host (calculated from frame header)
 - Receive Buffer: Updated from received frame
- Payload Length Configured
 - Length of data section (2-byte words) as configured by the Host
- Payload Length Received
 - Length of payload segment (2-byte words) stored from received frame

(3) Header section 3 (word 2)

Write access via FLXAnFRWRHS3, read access via FLXAnFRRDHS3:

- Data Pointer
 - Pointer to the beginning of the corresponding data section in the data partition

Read access via FLXAnFRRDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count
 - Cycle count from received frame
- RCI
 - Received on Channel Indicator

- SFI
 - Startup Frame Indicator
- SYN
 - Sync Frame Indicator
- NFI
 - Null Frame Indicator
- PPI
 - Payload Preamble Indicator
- RES
 - Reserved bit

(4) Message Buffer Status FLXAnFRMBS (word 3)

Read access via FLXAnFRMBS, updated by the CC at the end of the configured slot.

- VFRA
 - Valid Frame Received on channel A
- VFRB
 - Valid Frame Received on channel B
- SEOA
 - Syntax Error Observed on channel A
- SEOB
 - Syntax Error Observed on channel B
- CEOA
 - Content Error Observed on channel A
- CEOB
 - Content Error Observed on channel B
- SVOA
 - Slot boundary Violation Observed on channel A
- SVOB
 - Slot boundary Violation Observed on channel B
- TCIA
 - Transmission Conflict Indication channel A
- TCIB
 - Transmission Conflict Indication channel B
- ESA
 - Empty Slot Channel A

- ESB
 - Empty Slot Channel B
- MLST
 - Message LoST
- FTA
 - Frame Transmitted on Channel A
- FTB
 - Frame Transmitted on Channel B
- Cycle Count Status
 - Actual cycle count when status was updated
- RCIS
 - Received on Channel Indicator Status
- SFIS
 - Startup Frame Indicator Status
- SYNS
 - Sync Frame Indicator Status
- NFIS
 - Null Frame Indicator Status
- PPIS
 - Payload Preamble Indicator Status
- RESS
 - Reserved bit Status

18.3.13.2 Data Partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception / transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the user has to assure that the data pointers point to addresses within the data partition. **Table 18.107** below shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer’s data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32bit word are unused (see **Table 18.107** below).

Table 18.107 Example for Structure of the Data Partition in the Message RAM

| Bit Word | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-------------|----|----|----|---------------|----|----|----|---------------|----|----|----|---------------|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| ... | unused | | | | unused | | | | unused | | | | unused | | | | ... | | | | | | | | | | | | | | | |
| ... | unused | | | | unused | | | | unused | | | | unused | | | | ... | | | | | | | | | | | | | | | |
| ... | MBn Data3 | | | | MBn Data2 | | | | MBn Data1 | | | | MBn Data0 | | | | ... | | | | | | | | | | | | | | | |
| ... | ... | | | | ... | | | | ... | | | | ... | | | | ... | | | | | | | | | | | | | | | |
| ... | MBn Data(m) | | | | MBn Data(m-1) | | | | MBn Data(m-2) | | | | MBn Data(m-3) | | | | ... | | | | | | | | | | | | | | | |
| ... | ... | | | | ... | | | | ... | | | | ... | | | | ... | | | | | | | | | | | | | | | |
| ... | ... | | | | ... | | | | ... | | | | ... | | | | ... | | | | | | | | | | | | | | | |
| ... | MB1 Data3 | | | | MB1 Data2 | | | | MB1 Data1 | | | | MB1 Data0 | | | | ... | | | | | | | | | | | | | | | |
| ... | ... | | | | ... | | | | ... | | | | ... | | | | ... | | | | | | | | | | | | | | | |
| 2046 | MB0 Data3 | | | | MB0 Data2 | | | | MB0 Data1 | | | | MB0 Data0 | | | | ... | | | | | | | | | | | | | | | |
| 2047 | unused | | | | unused | | | | MB0 Data5 | | | | MB0 Data4 | | | | ... | | | | | | | | | | | | | | | |

18.3.13.3 Message Data Integrity Check

There is a data integrity checking mechanism implemented in the FlexRay core to assure the integrity of the data stored in the related RAM. Each RAM has a checksum generator / checker attached as shown in **Figure 18.19**.

When data is written to a RAM, the local checksum generator generates the checksum. The checksum is stored together with the respective data word. The checksum is checked each time a data word is read from a RAM.

If a checksum error is detected, the respective access error flag is set. The access error flags (bits AMR, ATBF1, and ATBF2) and the faulty message buffer indicators (bits FMBD, MFMB, and FMB) are located in the FlexRay Message Handler Status register (FLXAnFRMHDS). These single access error flags control the error interrupt flag (bit FLXAnFREIR.AERR).

Figure 18.19 shows the data paths between the Input Buffer, TBF and Message RAM.

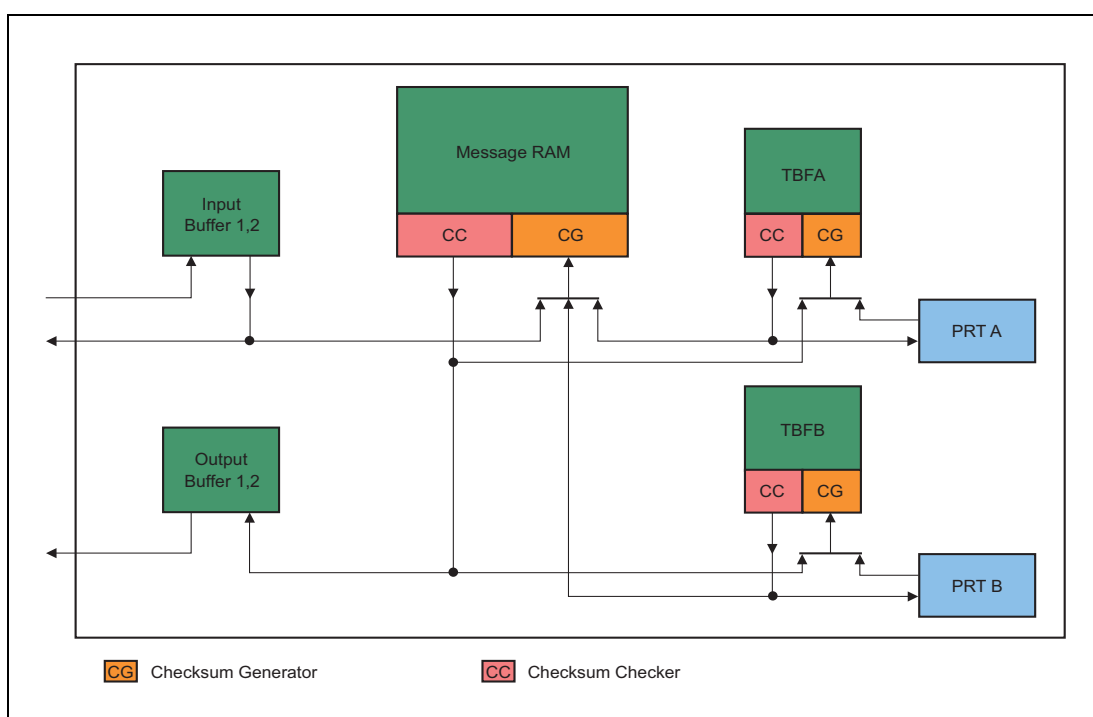


Figure 18.19 Checksum Generation and Check

When an access error has been detected the following actions will be performed:

In all cases:

- The respective access error flag in FLXAnFRMHDS register is set
- The access error flag FLXAnFREIR.AERR is set and, if enabled, a module interrupt to the Host will be generated.

Additionally in specific cases:**(1) Access error during data transfer from Input Buffer 1, 2 to Message RAM when reading header section of respective message buffer from Message RAM:**

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.
- Transmit buffer: Transmission request for the respective message buffer is not set.

(2) Access error during scan of header sections in Message RAM:

- FLXAnFRMHDS.AMR is set.
- Bit FLXAnFRMHDS.FMBD is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- Ignore message buffer (message buffer is skipped).

(3) Access error during data transfer from Message RAM to TBFA and TBFB:

- FLXAnFRMHDS.AMR is set.
- Bit FLXAnFRMHDS.FMBD is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero.

(4) Access error during data transfer from TBFA and TBFB to Message RAM when reading header section of respective message buffer from Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.

(5) Access error during data transfer from Message RAM to Output Buffer:

- The FLXAnFRMHDS.AMR bit is set to 1.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.

(6) Access error during a data transfer from TBFA and TBFB to Protocol Controller 1, 2:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- Frames already in transmission are invalidated by setting the frame CRC to zero.

(7) Access error during data transfer from TBFA and TBFB to Message RAM when reading TBFA and TBFB:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.

(8) Access error during data read of TBFA and TBFB:

When an access error occurs while the Message Handler read a frame with network management information (PPI = 1) from the TBFA and TBFB, the corresponding network management vector registers FLXAnFRNMV1 to 3 are not updated from this frame.

18.3.13.4 Host Handling of Access Errors

Access error caused by bit flips can be fixed by:

(1) Self-healing

Access errors located in the Data Section of Message RAM, TBFA or TBFB are overwritten with the next write access to the disturbed bit(s) caused by Host access or by FlexRay communication.

(2) CLEAR_RAM Command

The POC command CLEAR_RAM initializes the message RAM to zero, when called in the DEFAULT_CONFIG or CONFIG state.

(3) Temporary Unlocking of Header Section

An access error in the header section of a locked message buffer can be fixed by a transfer from the Input Buffer to the locked buffer Header Section. For this transfer, the write access to the FLXAnFRIBCR register (specifying the message buffer number) must be immediately preceded by the unlock sequence normally used to leave CONFIG state (see **Section 18.2.3.1, FLXAnFRLCK — FlexRay Lock Register**).

For that single transfer the respective message buffer header is unlocked, regardless whether it belongs to the FIFO or whether its locking is controlled by FLXAnFRMRC.SEC, and will be updated with new data.

18.3.14 Interrupts

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the CC, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the Host to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many interrupts can cause the Host to miss deadlines required for the application. Therefore the CC supports enable / disable controls for each individual interrupt source separately.

An interrupt may be triggered when

- An error was detected
- A status flag is set to 1
- A timer reaches a preconfigured value
- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A message transfer from the Local RAM/Global RAM to Message RAM or from Message RAM to Local RAM/Global RAM has completed
- A stop watch event occurred

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The Host has access to the actual status and error information by reading registers FLXAnFREIR, FLXAnFRSIR, FLXAnFROS, FLXAnFROTS and FLXAnFRITS.

The general purpose interrupt lines to the Host, FlexRay Interrupt 0, and FlexRay Interrupt 1 are controlled by the interrupts enabled by the FLXAnFREIES and FLXAnFRSIES registers. In addition each of the two interrupt lines can be enabled / disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

The input data transfer interrupt lines to the Host, FlexRay input queue empty interrupt, FlexRay input queue full interrupt, are controlled by the enabled interrupts in FLXAnFRITS. In addition, each of the input data transfer interrupts can be enabled or disabled separately by setting the related bits in FLXAnFRITC.

The output data transfer interrupt lines to the Host, FlexRay FIFO transfer warning interrupt, FlexRay output transfer warning interrupt, FlexRay FIFO transfer interrupt, FlexRay output transfer interrupt, are controlled by the enabled interrupts in FLXAnFROTS. In addition each of the output data transfer interrupts can be enabled or disabled separately by setting the related bits in FLXAnFROTC.

The three timer interrupts lines to the Host are controlled by the enabled interrupts in FLXAnFROS. In addition each of the interrupt lines can be enabled or disabled separately by setting bit FLXAnFROC.T0IE, FLXAnFROC.T1IE and FLXAnFROC.T2IE.

When a transfer between IBF/OBF and the Message RAM has completed bit FLXAnFRSIR.TIBC or FLXAnFRSIR.TOBC is set to 1.

An stop watch event is generated by the stpwt_extfxr pin input.

18.3.15 Assignment of FlexRay Configuration Parameters

Table 18.108 FlexRay configuration parameters (1/2)

| Parameter | Bit (field) |
|-----------------------------------|-------------------------|
| pKeySlotUsedForStartup | FLXAnFRSUCC1.TXST |
| pKeySlotUsedForSync | FLXAnFRSUCC1.TXSY |
| gColdStartAttempts | FLXAnFRSUCC1.CSA[4:0] |
| pAllowPassiveToActive | FLXAnFRSUCC1.PTA[4:0] |
| pWakeupChannel | FLXAnFRSUCC1.WUCS |
| pSingleSlotEnabled | FLXAnFRSUCC1.TSM |
| pAllowHaltDueToClock | FLXAnFRSUCC1.HCSE |
| pChannels | FLXAnFRSUCC1.CCHA, CCHB |
| pdListenTimeOut | FLXAnFRSUCC2.LT[20:0] |
| gListenNoise | FLXAnFRSUCC2.LTN[3:0] |
| gMaxWithoutClockCorrectionPassive | FLXAnFRSUCC3.WCP[3:0] |
| gMaxWithoutClockCorrectionFatal | FLXAnFRSUCC3.WCF[3:0] |
| gNetworkManagementVectorLength | FLXAnFRNEMC.NML[3:0] |
| gdTSSTransmitter | FLXAnFRPRTC1.TSST[3:0] |
| gdCASRxLowMax | FLXAnFRPRTC1.CASM[6:0] |
| gdSampleClockPeriod | FLXAnFRPRTC1.BRP[1:0] |
| pSamplesPerMicrotick | FLXAnFRPRTC1.BRP[1:0] |
| gdWakeupSymbolRxWindow | FLXAnFRPRTC1.RXW[8:0] |
| pWakeupPattern | FLXAnFRPRTC1.RWP[5:0] |
| gdWakeupSymbolRxIdle | FLXAnFRPRTC2.RXI[5:0] |
| gdWakeupSymbolRxLow | FLXAnFRPRTC2.RXL[5:0] |
| gdWakeupSymbolTxIdle | FLXAnFRPRTC2.TXI[5:0] |
| gdWakeupSymbolTxLow | FLXAnFRPRTC2.TXL[5:0] |
| gPayloadLengthStatic | FLXAnFRMHDC.SFDL[6:0] |
| pLatestTx | FLXAnFRMHDC.SLT[12:0] |
| pMicroPerCycle | FLXAnFRGTUC1.UT[19:0] |
| gMacroPerCycle | FLXAnFRGTUC2.MPC[13:0] |
| gSyncNodeMax | FLXAnFRGTUC2.SNM[3:0] |
| pMicroInitialOffset[A] | FLXAnFRGTUC3.UIOA[7:0] |
| pMicroInitialOffset[B] | FLXAnFRGTUC3.UIOB[7:0] |
| pMacroInitialOffset[A] | FLXAnFRGTUC3.MIOA[6:0] |
| pMacroInitialOffset[B] | FLXAnFRGTUC3.MIOB[6:0] |
| gdNIT | FLXAnFRGTUC4.NIT[13:0] |
| gOffsetCorrectionStart | FLXAnFRGTUC4.OCS[13:0] |
| pDelayCompensation[A] | FLXAnFRGTUC5.DCA[7:0] |
| pDelayCompensation[B] | FLXAnFRGTUC5.DCB[7:0] |
| pClusterDriftDamping | FLXAnFRGTUC5.CDD[4:0] |
| pDecodingCorrection | FLXAnFRGTUC5.DEC[7:0] |
| pdAcceptedStartupRange | FLXAnFRGTUC6.ASR[10:0] |
| pdMaxDrift | FLXAnFRGTUC6.MOD[10:0] |
| gdStaticSlot | FLXAnFRGTUC7.SSL[9:0] |
| gNumberOfStaticSlots | FLXAnFRGTUC7.NSS[9:0] |

Table 18.108 FlexRay configuration parameters (2/2)

| Parameter | Bit (field) |
|-----------------------------|-------------------------|
| gdMinislot | FLXAnFRGTUC8.MSL[5:0] |
| gNumberOfMinislots | FLXAnFRGTUC8.NMS[12:0] |
| gdActionPointOffset | FLXAnFRGTUC9.APO[5:0] |
| gdMinislotActionPointOffset | FLXAnFRGTUC9.MAPO[4:0] |
| gdDynamicSlotIdlePhase | FLXAnFRGTUC9.DSI[1:0] |
| pOffsetCorrectionOut | FLXAnFRGTUC10.MOC[13:0] |
| pRateCorrectionOut | FLXAnFRGTUC10.MRC[10:0] |
| pExternOffsetCorrection | FLXAnFRGTUC11.EOC[2:0] |
| pExternRateCorrection | FLXAnFRGTUC11.ERC[2:0] |

18.3.16 Usage of Data Transfer

The data transfer function is for the transfer of FlexRay messages between the FlexRay module's internal message RAM and areas of user RAM such as Local RAM/Global RAM.

Data transfer from an area of user RAM to the internal message RAM (input) is started by software. Input transfer is used to set up a message buffer or to update data for transmission.

Data transfer from the internal message RAM to an area of user RAM (output) starts in response to the following events.

- Data are stored in the reception message buffer or the internal FIFO buffer of the FlexRay module.
- The state of a slot changes.
- The user issues a transfer request.

An area of the user RAM needs to be secured for the data to be transferred. The data are transferred in a defined structure.

Access to the input and output buffers by the CPU is prohibited while this function is in use.

18.3.16.1 Input Data Transfer

When the automatic input data transfer function is enabled, committed input data structures are transferred from the Local RAM/Global RAM to the FlexRay internal message RAM with minimum CPU support.

(1) Activation and deactivation

The input data transfer function should be activated before usage. The activation of the input transfer handler initializes the input queue put index (FLXAnFRITS.IPIDX[6:0]) and get index (FLXAnFRITS.IGIDX[6:0]) to zero. Also the interrupt status bits in the FLXAnFRITS register (IQEIS and IQFIS) are set to 0.

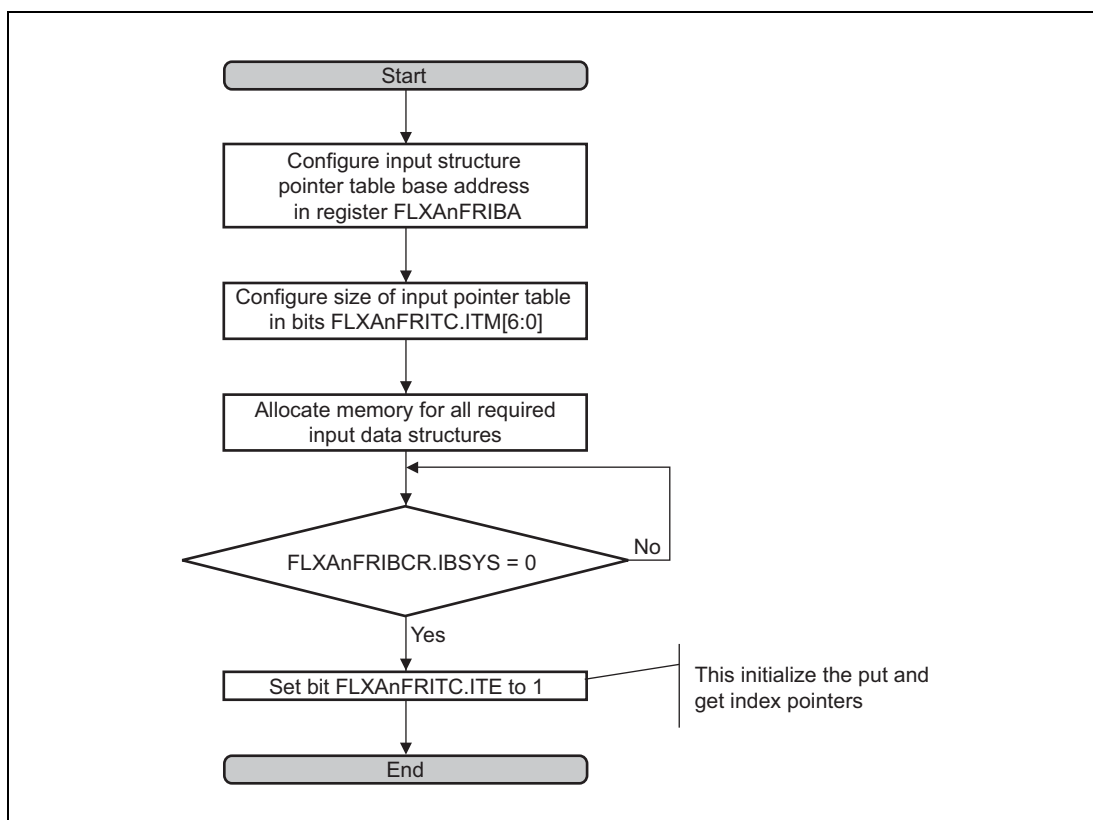


Figure 18.20 Input Transfer Enable Processing Flow

A deactivation request of the input transfer function can be made at any time. The input queue put index and the input queue status are maintained independently from the input transfer function state.

Before the transfer function gets disabled (bit FLXAnFRITS.ITS = 0), user requested input transfers and all committed input transfers will be completed.

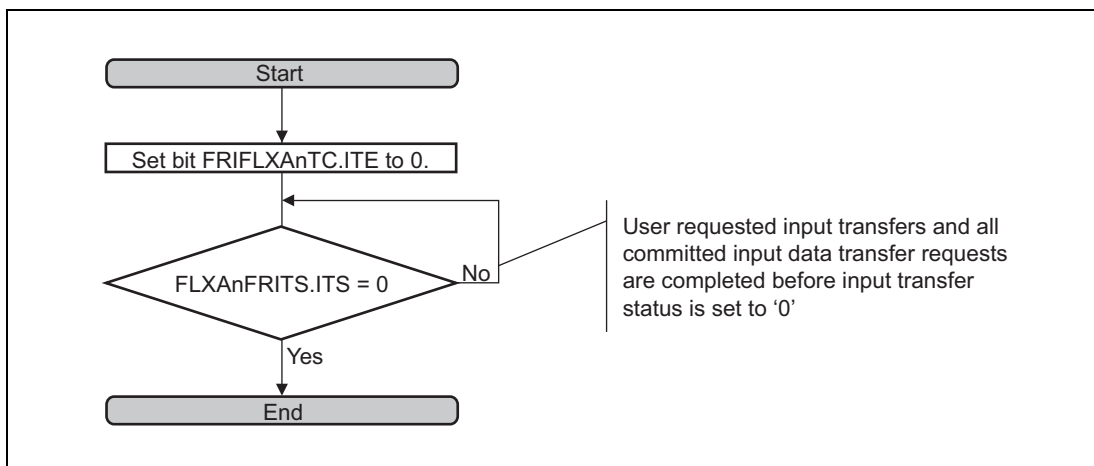


Figure 18.21 Input Transfer Disable Processing Flow

(2) Input data structure

The application has to reserve a location in the Local RAM/Global RAM to provide the content for message buffer configuration (input data structure).

The location of this input data structure needs to be defined by an input data structure pointer also located in the Local RAM/Global RAM.

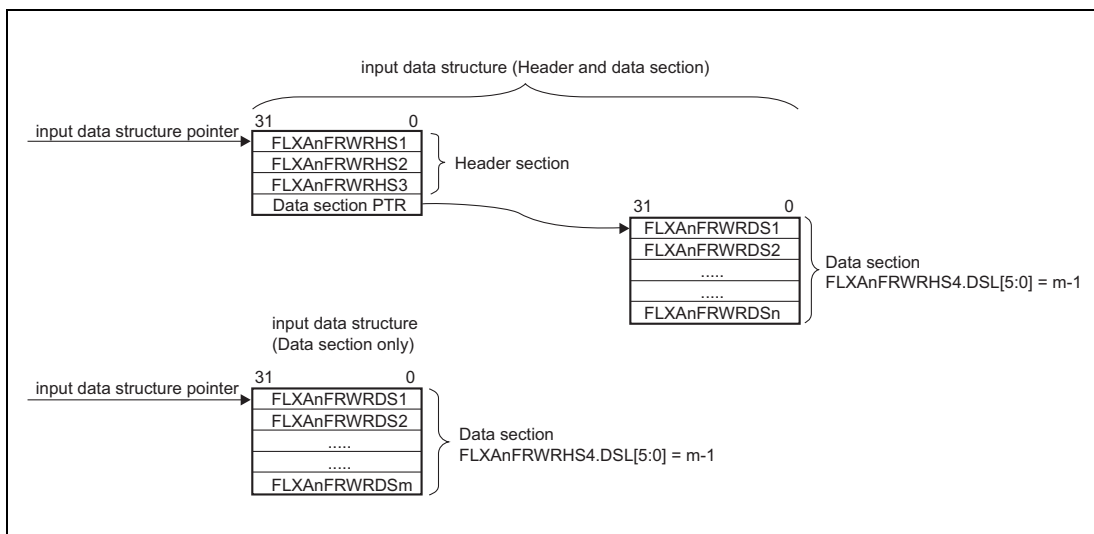


Figure 18.22 Input data structure

In general the input data structure consists of two sections, the header and the data section.

The header section consists of FLXAnFRWRHS1, FLXAnFRWRHS2, FLXAnFRWRHS3 and the data section pointer.

For bit alignment and bit function in the header section, see **Section 18.3.13.1, Header Partition**.

Depending on the settings in the control field (FLXAnFRWRHS4) located in the input pointer table, the data structure pointer is a reference to the address of FLXAnFRWRHS1 or FLXAnFRWRDS1. The data structure pointer has to be aligned to a 32 bit address.

If bit FLXAnFRWRHS4.LHS is set to 1, it is required to provide a valid header section. In this case FLXAnFRWRDS1 is the first element of the data structure.

If bit FLXAnFRWRHS4.LHS is set to 0, a header section is not required. In this case FLXAnFRWRDS1 is the first element of the data structure.

If bit FLXAnFRWRHS4.LDS is set to 1, it is required to provide a valid data section. The pointer to the data section is a reference to the address of the first payload long word (FLXAnFRWRDS1) and has to be aligned to a 32 bit address.

If bit FLXAnFRWRHS4.LDS is set to 0, a data section is not required. The data section pointer is not evaluated by the input handler.

The length of the data section and the size to be allocated in the Local RAM/Global RAM depends on the configuration of bits FLXAnFRWRHS4.DSL[5:0].

For the transfer into the FlexRay core internal message RAM the number of 16 bit words configured by bits FLXAnFRWRHS2.PLC[6:0] is used. The application has to ensure, that a proper number of data words is provided in the Local RAM/Global RAM. In case the buffer is configured by bits FLXAnFRWRHS2.PLC[6:0] to hold an odd payload length, the application has to write zero to the last 16 bit of the payload section to ensure that the padding data is all zero.

(3) Input pointer table

To transfer data from the input data structures located in the Local RAM/Global RAM to the FlexRay internal message RAM the related input data structure pointer and control field needs to be added to the input pointer table which is located in the Local RAM/Global RAM.

The location of the first element of this table is identified by the input pointer table base address (FLXAnFRIBA.ITA[31:0]). This base address has to be aligned to a 32 bit address.

The maximum number of input requests that can be queued is defined by the Input queue Table Max register (FLXAnFRITC.ITM[6:0]).

Each Input pointer table entry requires two long words. The required address range of the input pointer table for the queued transfer requests can be calculated by

$$\text{Input pointer table size (byte)} = (((\text{FLXAnFRITC.ITM}[6:0] + 1) \times 2) \times 4)$$

Equation 1

The input pointer entry for the user requested input transfer should be added after the end of the input pointer table.

The pointer table index related to this entry and hence the number to be written to FLXAnFRUIR.UIDX[7:0], is FLXAnFRITC.ITM[6:0] + 1. The address in the input pointer table related to the user requested input transfer (user input address) can be calculated by

$$\text{User input address} = \text{FLXAnFRIBA.ITA}[31:0] + \text{Input pointer table size}$$

Equation 2

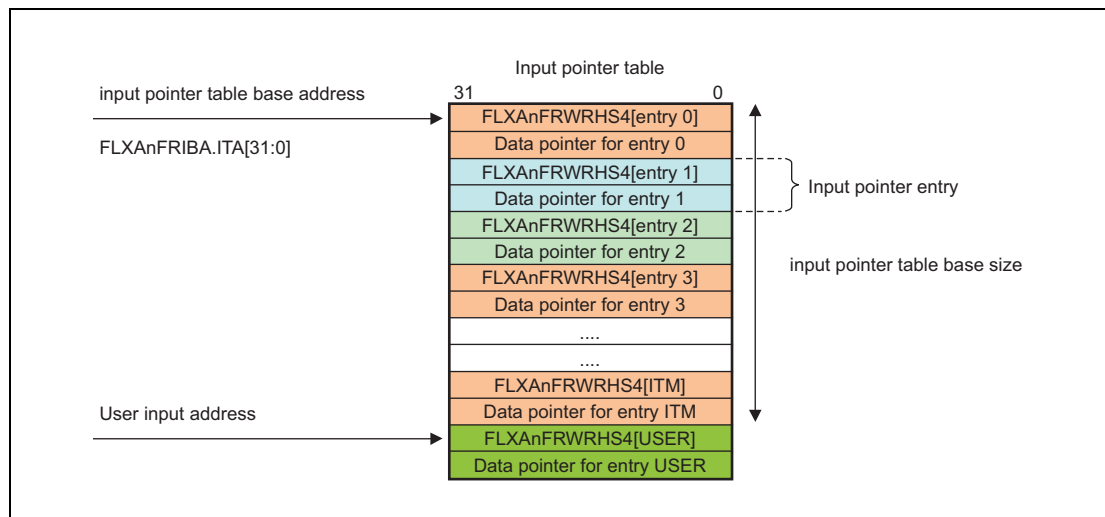


Figure 18.23 Input pointer table

The input pointer table holds the control field FLXAnFRWRHS4 and the pointers to the Local RAM/Global RAM location where the message buffer content (header section and/or data section) is stored.

The application has to write FLXAnFRWRHS4 and the input data structure pointer at the addresses in the input pointer table related to the put index position before a transfer request is initiated.

FLXAnFRWRHS4:

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|-----|-----|-----|-----|----|------------|----------|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | DSL[5:0] | | | | | |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | INV | STR | LDS | LHS | — | IMBNR[6:0] | | | | | | |
| Value after reset | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

Table 18.109 FLXAnFRWRHS4 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 22 | Reserved | When read, an undefined value is read. When writing, always write 0. |
| 21 to 16 | DSL[5:0] | Data Section Length Bit Specifies the length of the data section in terms of 32 bit values. |
| 15 to 12 | Reserved | When read, an undefined value is read. When writing, always write 0. |
| 11 | INV | Invalidate entry Bit 0: The data structure is valid and will be transferred to the FlexRay internal message RAM. 1: The data structure is invalid. FlexRay internal message RAM is not updated using this input pointer entry. |
| 10 | STR | Set transmission request Bit 0: The bit FLXAnFRTXRQm.TXRp for the message buffer selected by the bits IMBNR[6:0] is set to 0. No data from this message buffer is transmitted. 1: The bit FLXAnFRTXRQm.TXRp for the message buffer selected by the bits IMBNR[6:0] is set to 1 to release the message buffer for transmission. The application should not set the bit STR to 1 for receive buffers. |
| 9 | LDS | Load data section Bit 0: No update of data section. 1: Data section for the message buffer selected by the bits IMBNR[6:0] is updated. |
| 8 | LHS | Load header section Bit 0: No update of header section. 1: Header section for the message buffer selected by the bits IMBNR[6:0] is updated. |
| 7 | Reserved | When read, an undefined value is read. When writing, always write 0. |
| 6 to 0 | IMBNR[6:0] | Message buffer number to be updated Bit Selects the target message buffer number in the FlexRay internal message RAM for transfer |

Note that the LHS bit should not be set for protected message buffers.

The bit LDS defines if the data section of the message buffer selected by the bits IMBNR[6:0] should be updated.

If LDS is set to 1 (DSL[5:0] + 1) 32 bit words of payload data are transferred from the Local RAM/ Global RAM to the message buffer selected by the bits IMBNR[6:0].

If LDS is set to 0 no payload data is transferred from the Local RAM/Global RAM.

Note that the payload transferred is independent from the configured payload length (bits FLXAnFRWRHS2.PLC[6:0]).

The bit INV can be used to invalidate a committed data structure. This bit should be only used to cancel the transfer of committed data structures when the input queue is halted (see **Section 18.3.16.1 (5), Transfer function of dedicated message buffers**).

When this bit is set to 1 the message buffer number (IMBNR[6:0]) is not updated. When the bit is set to 0 the message buffer number (IMBNR[6:0]) is updated.

(4) Transfer function of input data structure

To use the input data structure transfer function the input transfer has to be activated (see **Section 18.3.16.1 (1), Activation and deactivation**). The activation process requires the setup of the input pointer table (see **Section 18.3.16.1 (3), Input pointer table**) in order to specify the source location (input data structures) for the data structures to be transferred. When the input transfer gets enabled the get index pointer is initialized to zero.

All FlexRay internal message buffers can be updated using the input transfer queue which is built in the input pointer table. The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. For that purpose the application has to maintain a put index for the input pointer table that indicates where the pointer has to be written to.

To commit this table entry to the input handler, the application has to write the target message buffer number to the input queue control register (bits FLXAnFRIQC.IMBNR[6:0]). Afterwards the application has to increment the application internal put index.

By writing to the input queue control register the data available bits (FLXAnFRDAm.DA[31:0]) are automatically set to 1. The input transfer handler also maintains the put index pointer in the status register (FLXAnFRITS.IPIDX[6:0]).

In case the input queue gets full (number of queued input transfer requests is equal to the input queue table size) FLXAnFRITS.IQFP and FLXAnFRITS.IQFIS are set to 1. The input queue full condition pending flag (FLXAnFRITS.IQFP) changes from 1 to 0 when there are entries in the input queue available, whereby the input queue full interrupt status flag (FLXAnFRITS.IQFIS) needs to be cleared by the application.

The application should not make any further write access to bits FLXAnFRIQC.IMBNR[6:0] as long as bit FLXAnFRITS.IQFP is 1.

In case the input queue gets empty (number of queued input transfer requests changes to zero) FLXAnFRITS.IQEIS is set to 1. The input queue empty interrupt status flag (FLXAnFRITS.IQEIS) needs to be cleared by the application.

The transfer of the input data structures to the FlexRay message RAM is controlled by a get index pointer which is handled inside the FlexRay module and flagged in bits FLXAnFRITS.IGIDX[6:0]. Note that the index is referring to the input entry and not the address offset in the input pointer table.

If the input queue is not empty, the transfer handler reads out the input pointer table entry of the transfer queue and starts the transfer of the input data structure from the address the input pointer is referring to. When all required data words are transferred to the FlexRay module, the data available flag for the transferred message buffer number is set to 0 and the get index in the transfer handler is incremented by one.

In case of an invalidated data structure (see **Section 18.3.16.1 (5), Halting the input queue**) no FlexRay internal message buffer is updated and the related data available flag is automatically set to 0. The change of the data available flag can be used to confirm the cancellation a transmit request.

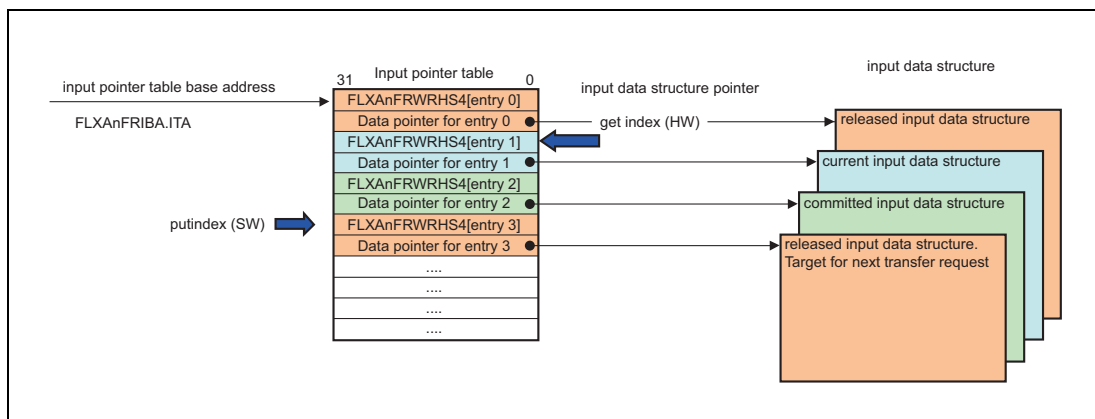


Figure 18.24 Input Pointer Table

Receive message buffers can be also configured using the input data transfer by setting up the required header sections and mark only the header section (FLXAnFRWRHS4.LDS = 0, FLXAnFRWRHS4.LHS = 1) to be updated in the FlexRay module.

(5) Halting the input queue

Committed data structures cannot be removed, but can be invalidated or updated when the input queue is halted.

To cancel data structures already committed to the input queue, the queue can be halted by writing a 1 to FLXAnFRITC.IQHR.

After the ongoing input transfer has been completed the queue is halted and FLXAnFRITS.IQH changes from 0 to 1.

To invalidate an entry of the input queue FLXAnFRWRHS4.INV has to be set to 1. All other bits in FRWRHS4 should be unchanged.

The following processing flow shall be used to analyze whether a committed message has been already transferred to the FlexRay internal message RAM or not.

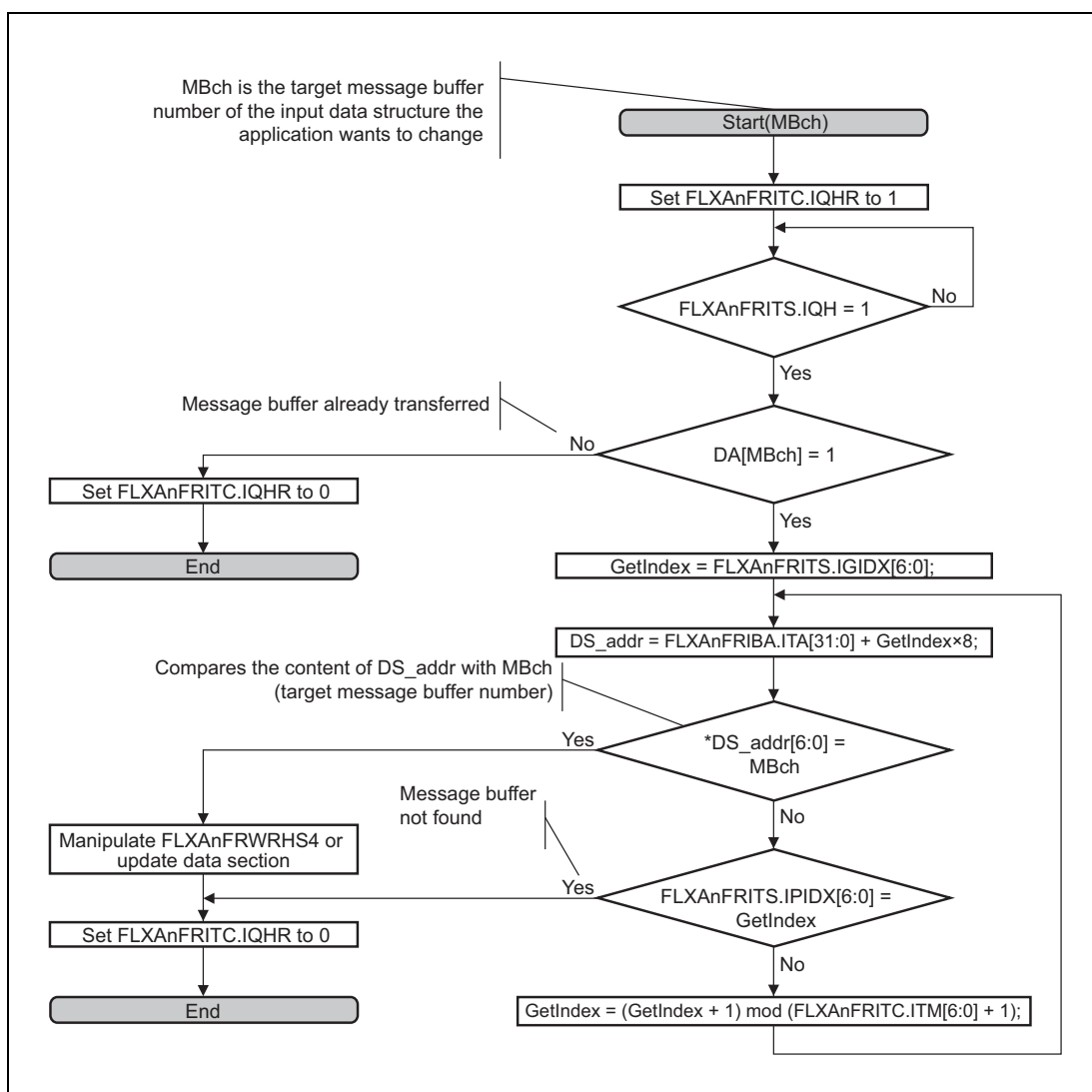


Figure 18.25 Input Table Analysis

In case the message buffer was already transferred to the FlexRay internal message RAM, the user input transfer request can be used to bypass the actual queue and update the required message buffer (see **Section 18.3.16.1 (6), Transfer function of user requested input transfers**).

(6) Transfer function of user requested input transfers

To use this function the input transfer has to be activated (see **Section 18.3.16.1 (1), Activation and deactivation**).

The application is capable, by using FLXAnFRUIR.UIDX[7:0], to request a transfer of an input data structure. The user input transfer request is serviced first.

The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. The table entry for the user input transfer request should be added after the end of the input pointer table (see **Section 18.3.16.1 (3), Input pointer table**).

To commit this table entry to the input handler, the application has to write the index (FLXAnFRITC.ITM[6:0] +1) to the user input transfer request register (FLXAnFRUIR.UIDX[7:0]).

By writing to the user input transfer request register, the user input transfer request pending flag (FLXAnFRITS.UIRP) is automatically set to 1.

As long this flag is 1 the application should not make any further user input transfer requests.

The user input transfer request pending flag (FLXAnFRITS.UIRP) changes from 1 to 0 when the requested input transfer is completed. As next the pending transfers are processed.

18.3.16.2 Output Data Transfer

When the output data transfer function is enabled, received messages (either in dedicated message buffers or in the FlexRay receive FIFO) are transferred to the Local RAM/Global RAM by the output data handler. The output data handler can also transfer the message buffer content to the Local RAM/Global RAM on application request. When enabled the output handler is also capable to initiate a transfer when the message buffer status has changed.

(1) Activation and deactivation

The output data transfer function should be activated before usage. The activation of the output transfer handler will initialize the FIFO put and get index pointer and FIFO fill level (FLXAnFROTS.FGIDX[4:0] and FLXAnFROTS.FFL[5:0]) to zero, set the bits FLXAnFROTS.FDA, FLXAnFROTS.OWP, FLXAnFROTS.FWP and FLXAnFROTS.UORP to 0. Also the interrupt status flags FLXAnFROTS.OTIS, FLXAnFROTS.FIS, FLXAnFROTS.OWIS and FLXAnFROTS.FWIS) are set to 0.

The activation has no influence to the data available flags (FLXAnFRDAm.DA[31:0]) which are related to the dedicated buffers; these flags have to be cleared by the application.

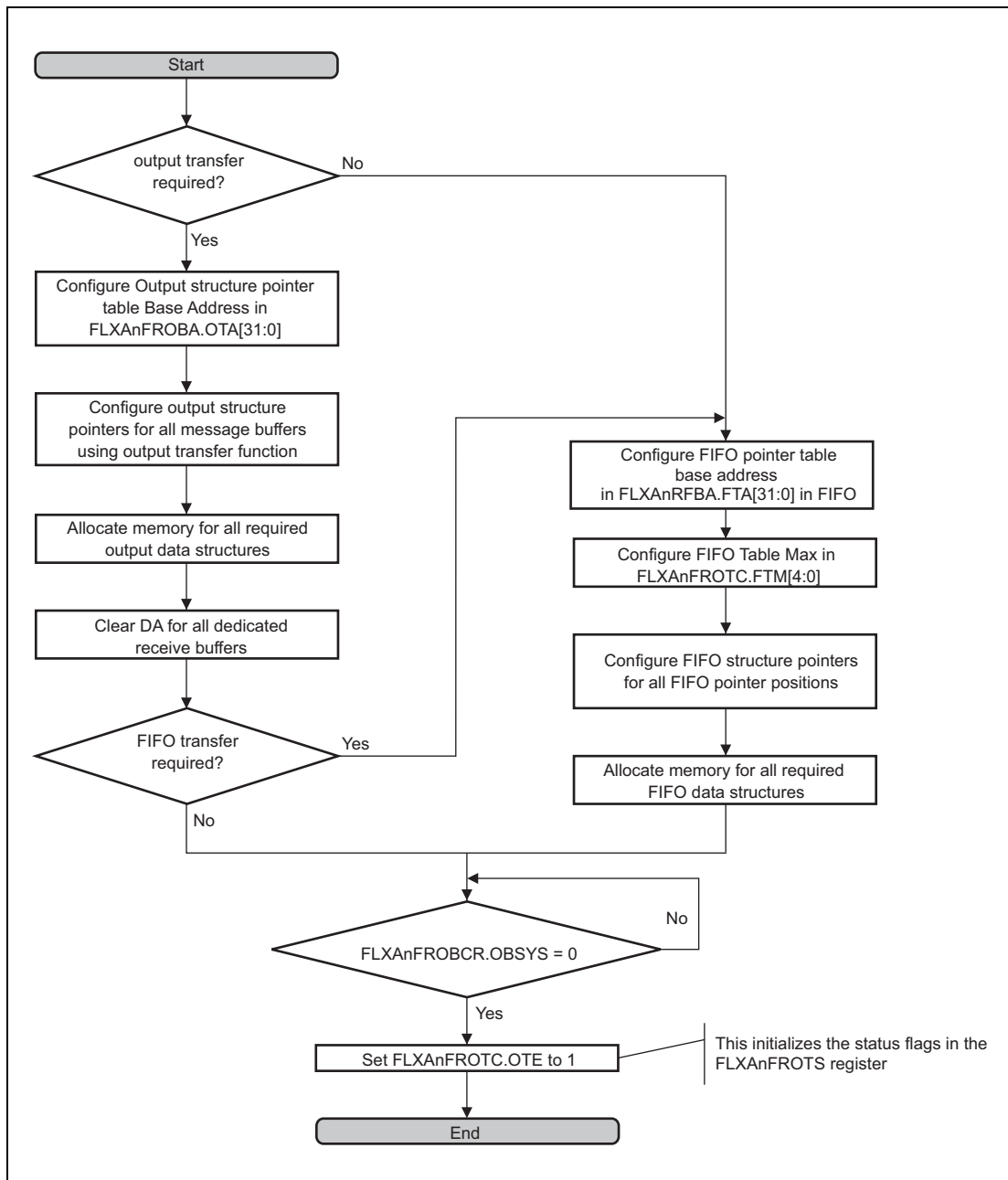


Figure 18.26 Output Transfer Enable Processing Flow

A deactivation request of the output data transfer function can be made at any time. An ongoing transfer will be completed and the completion of this transfer will be flagged. During this time FLXAnFROTS.OTS remains 1.

When FLXAnFROTS.OTS changes from 1 to 0, the output transfer function is deactivated. The data available status flags and the FIFO get index are still retained when the output transfer function is disabled.

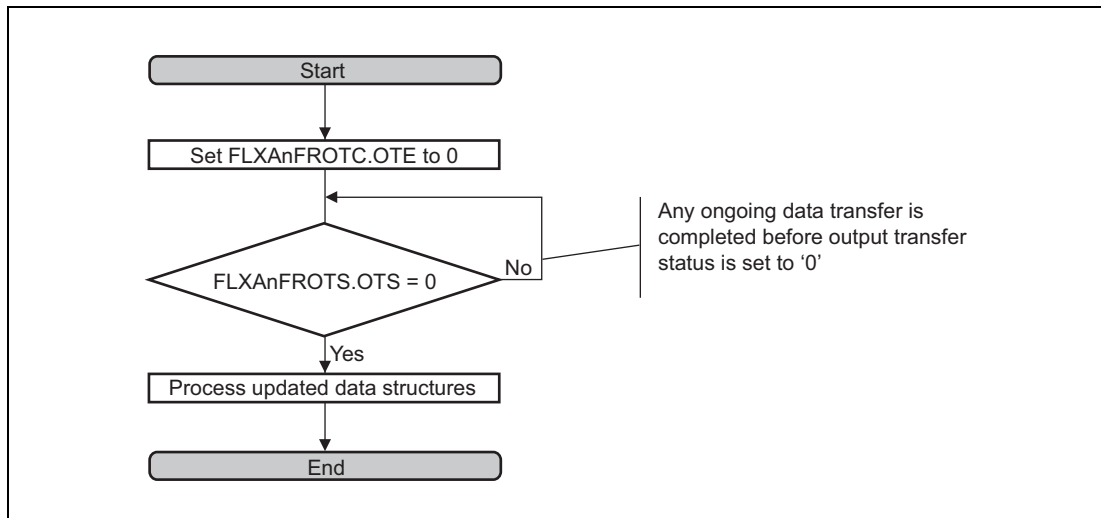


Figure 18.27 Output Transfer Disable Processing Flow

(2) Output transfer data structure

The data in the Local RAM/Global RAM is stored in an output data structure. The location of the output data structures are determined by output data structure pointers also located in the Local RAM/Global RAM. The output data structure and indexing is visualized in **Figure 18.28**.

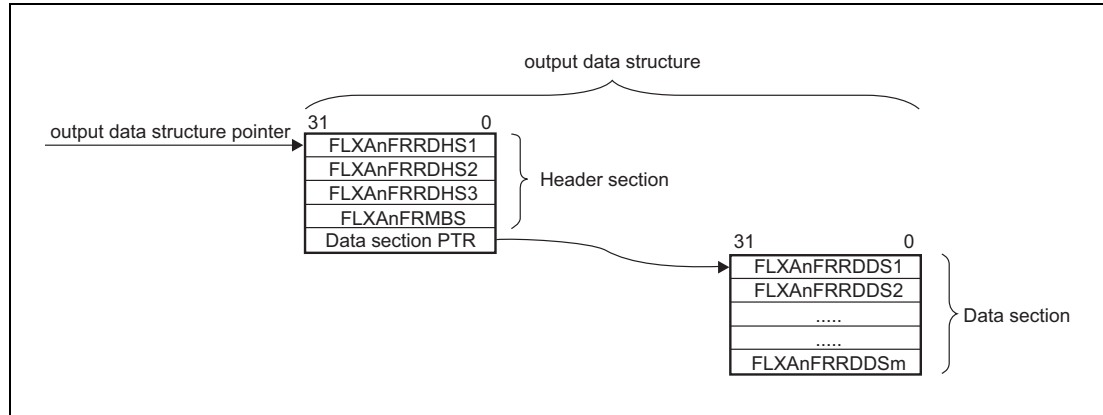


Figure 18.28 Output Data Structure

The output data structure consists of two sections, the header and data section. The header section consists of FLXAnFRRDHS1, FLXAnFRRDHS2, FLXAnFRRDHS3, FLXAnFRMBS and the data section pointer. FLXAnFRRDHS1 is the first element of the structure and has to be aligned to a 32 bit address. The data structure pointer is a reference to the address of FLXAnFRRDHS1. For information about the bit alignment and bit function within the header section refer to **Section 18.3.13.1, Header Partition**.

FLXAnFRRDDS1 is the first element of the data section. The data section pointer needs to be aligned with a 32-bit address corresponding to the FLXAnFRRDDS1 address.

The length of the data section as well as the total structure size to be allocated in the Local RAM/Global RAM depends on the configured payload length (bits FLXAnFRRDHS2.PLC[6:0]) of the related message buffer. In case the configured payload length is an odd number of words or the received payload length (bits FLXAnFRRDHS2.PLR[6:0]) is smaller than the configured payload length, the remaining data words in the Local RAM/Global RAM are unused and should not be used by the application.

The output data structure is identical for all three kinds of output transfers. In case only the header section is transferred the data section pointer is not evaluated by the output handler and the data section remains unchanged.

(3) Output pointer table

For the output data transfer function the application needs to setup an output pointer table in the Local RAM/Global RAM. The location of the first element of this table should be programmed into the output pointer table base address (bits FLXAnFROBA.OTA[6:0]). This base address has to be aligned to a 32 bit address.

The size of the output pointer table is defined by the maximum of: the last configured dedicated message buffer and the highest message buffer number which will be used for the user output transfer request.

The output pointer table holds pointers (output data structure pointers) to the Local RAM/Global RAM location where a memory space is reserved for the target message buffer content (header section and data section).

There is a fixed linear relationship between the address of the entries in the output pointer table and the number of the related message buffers (see **Figure 18.29**): the output pointer table starts with the entry for message buffer number 0 at the address configured in bits FLXAnFROBA.OTA[31:0] and continues in ascending order for each following message buffer number, by 32 bit aligned address (e.g. message buffer 1 at output pointer table address FLXAnFROBA.OTA[31:0] + 4, message buffer 2 at output pointer table address FLXAnFROBA.OTA[31:0] + 8, etc.) for all possible message buffers.

When a set of bit FLXAnFRNDATm.NDp is the only transfer condition (FLXAnFROTC.OTCS is set to 0), only message buffers configured as a dedicated receive buffer or that will be used for user output transfer requests need have valid pointer entries.

When a set of bit FLXAnFRNDATm.NDp or FLXAnFRMBSCm.MBCp is the transfer condition (FLXAnFROTC.OTCS is set to 1), all dedicated receive buffer and dedicated transmit buffers need to have valid pointer entries.

(4) FIFO output pointer table

The FlexRay module internal FIFO can be extended by a queued buffer structure in the Local RAM/Global RAM.

If the FlexRay module internal FIFO is used the application needs to setup the FIFO output pointer table. The location of the first element of this table is identified by the FIFO pointer table base address (bits FLXAnFRFBA.FTA[31:0]). This base address has to be aligned to a 32 bit address.

The size of the FIFO pointer table and hence the maximum number of messages that can be added to the queue, is defined by FIFO Table Max (bits FLXAnFROTC.FTM[4:0]).

The FIFO pointer table holds pointers (output data structure pointers) to the Local RAM/Global RAM location where a memory space is reserved the target message buffer content (header section and data section). For each table entry a data pointer shall be configured in this table.

(5) Transfer function of dedicated message buffers

To use this transfer function the output transfer has to be activated (see **Section 18.3.16.2 (1), Activation and deactivation**). The activation process requires to setup the output pointer table (see **Section 18.3.16.2 (3), Output pointer table**) in order to specify the destination location (output data structures) for the data to transfer. **Figure 18.29** shows how the output pointer table references the output data structures.

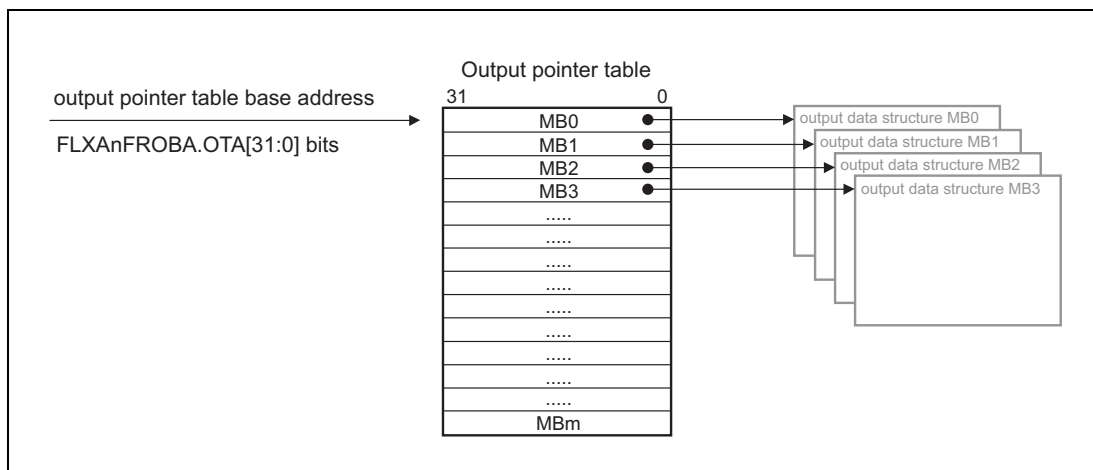


Figure 18.29 Output Data Structure and Indexing

With FLXAnFROTC.OTCS the output transfer condition can be selected between the 'New data only mode' and the 'New data and status changed mode'.

In the 'New data only mode' an output data transfer is initiated when a valid FlexRay data frame has been stored into a dedicated receive buffer which causes the related FLXAnFRNDATm.NDp flag to set. The FLXAnFRNDATm.NDp flag is automatically set to 0 during the transfer procedure. The header section is also transferred and hence the FLXAnFRMBSCm.MBCp flag is set to 0.

In the 'New data and status changed mode' an output data transfer is initiated as described in the 'New data only mode'. In addition an output data transfer is initiated when only the message buffer status has been changed which causes the related FLXAnFRMBSCm.MBCp flag to be set. In this case only the header section is transferred. The FLXAnFRMBSCm.MBCp flag is automatically set to 0 during the transfer procedure.

After transferring the message buffer data from the FlexRay internal message RAM to the output data structure the corresponding data available flag in the FLXAnFRDAm ($m = 0$ to 3) registers is set to 1. The update of the output data structure is also flagged by the setting of the output transfer interrupt status flag (FLXAnFROTS.OTIS).

As long as the data available flag remains 1 the corresponding output data structure will not be updated.

In the case

- the data available flag is 1 and a valid received message was stored or
- when FLXAnFROTC.OTCS is 1 and the message buffer status was updated,

the output transfer warning interrupt flag (FLXAnFROTS.OWIS) is set to 1 notifying the application that new data is available but the output data structure transfer cannot be processed. In addition FLXAnFROTS.OWP is set to 1 that continuously flags that status of the output transfer warning condition.

If a valid receive message in the FlexRay internal message RAM is overwritten by an additional receive message, the message lost flag (FLXAnFRMBS.MLST) is set to 1. This flag can be evaluated after the message buffer has been transferred into an output data structure.

Following sections are giving a guidance how output data structures can be handled.

(a) Data section copy method

One option is to copy the information from the output data structure to a different location of the Local RAM/Global RAM and then release the output data structure by clearing the related data available flag. The application should use the copied information for further processing.

(b) Data structure pointer method

A different option is to modify the output data structure pointer in the output pointer table and to release the output data structure by clearing the related data available flag. The changed output data pointer should refer to a free data structure. The application should use the old data structure for further processing.

(c) Data section pointer method

A third option is to modify the data section pointer in the output data structure and to release the output data structure by clearing the related data available flag. The changed data section pointer should refer to a free memory area. The application should use the old data section for further processing by forwarding the data section pointer.

(6) Transfer function of FIFO message buffers

To use this buffer transfer function the output transfer has to be activated (see **Section 18.3.16.2 (1), Activation and deactivation**). The activation process requires the setup of the FIFO pointer table (see **Section 18.3.16.2 (4), FIFO output pointer table**) in order to specify a location in the Local RAM/Global RAM reserved for the storage of the required output data structures.

A FIFO data transfer is initiated when a valid FlexRay data frame has been stored in the FlexRay internal FIFO.

After transfers from the internal FIFO to the output data structure, the FIFO interrupt status flag (FLXAnFROTS.FIS) and FIFO data available bit (FLXAnFROTS.FDA) are set to 1. The bit FLXAnFROTS.FIS can be used as an interrupt source. The bit FLXAnFROTS.FDA indicates that the FIFO is not empty.

The size of the output structure is up to the maximum number of entries in the FIFO table (as set by the FLXAnFROTC.FTM[4:0] bits).

The transfer to the extended FIFO buffer structure is controlled by index pointers. This put index is controlled by the FIFO transfer handler and is incremented after transferring a message to the output data structure.

The FIFO reception handler also maintains a get index which is indicated in FLXAnFROTS.FGIDX[4:0]. The value of this get index is known by the application by either reading the status or maintaining a software variable. The get index (the value after a reset is 00000_B) is incremented by one when the application releases the oldest entry of the FIFO queue by writing 1 to FLXAnFROTS.FDA. By comparing the put index and the get index the FIFO handler knows about the current fill level of the queued buffer structure.

The current FIFO fill level is flagged in FLXAnFROTS.FFL[5:0]. When FLXAnFROTS.FDA is 1, there is at least one entry in the FIFO queue.

In case the queued buffer structure in the Local RAM/Global RAM is full (FLXAnFROTS.FFL[5:0] = FLXAnFROTC.FTM[4:0] + 1), no further transfers are initialized, new messages remain in the FlexRay internal FIFO and the FIFO transfer warning interrupt status flag (FLXAnFROTS.FWIS) is set to 1.

In case the FlexRay internal FIFO structure becomes full, messages in the FlexRay internal FIFO structure may be overwritten. The related status flags and configuration registers of the FlexRay core module can be used to generate desired warning notifications.

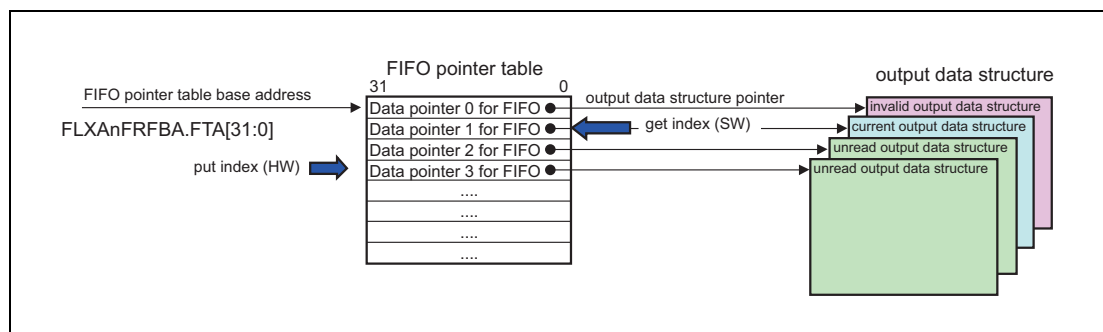


Figure 18.30 FIFO Pointer Table

(7) Transfer function of user output transfer requests

To use this transfer function the output transfer has to be activated (see **Section 18.3.16.2 (1), Activation and deactivation**). The activation process requires to setup the output pointer table (see **Section 18.3.16.2 (3), Output pointer table**) in order to specify the location in the Local RAM/ Global RAM reserved for the transfer of the data (output data structures).

The application is capable, by using FLXAnFRUOR.UMBNR[6:0], to request a transfer of dedicated message buffer to an output data structure. Except in CONFIG state, message buffers which are part of the FlexRay internal FIFO should not be requested.

The header section is always transferred to the output data structure. The transfer of the data section can be enabled by setting FLXAnFRUOR.URDS to 1. The selected message buffer content is stored in the output data structure location determined by the pointers in the output pointer table.

The data available status and transfer blocking by bits FLXAnFRDAm.DA[31:0] is also used for the user requested transfers. Therefore bits FLXAnFRDAm.DA[31:0] related to the requested buffer number (FLXAnFRUOR.UMBNR[6:0]) should be released before making the transfer request.

After writing to FLXAnFRUOR.UMBNR[6:0], the bit FLXAnFROTS.UORP is set to 1 to indicate that there is a pending user transfer request. When the transfer has been processed the bit FLXAnFROTS.UORP is set to 0, the bit FLXAnFROTS.OTIS is set to 1 and bits FLXAnFRDAm.DA[31:0] related to the requested buffer number (FLXAnFRUOR.UMBNR[6:0]) are set to 1.

User output transfer requests cannot be queued. The application should check the bit FLXAnFROTS.UORP before writing to FLXAnFRUOR.UMBNR[6:0].

User output transfer requests should not be made for message buffers which are pending in the input transfer queue.

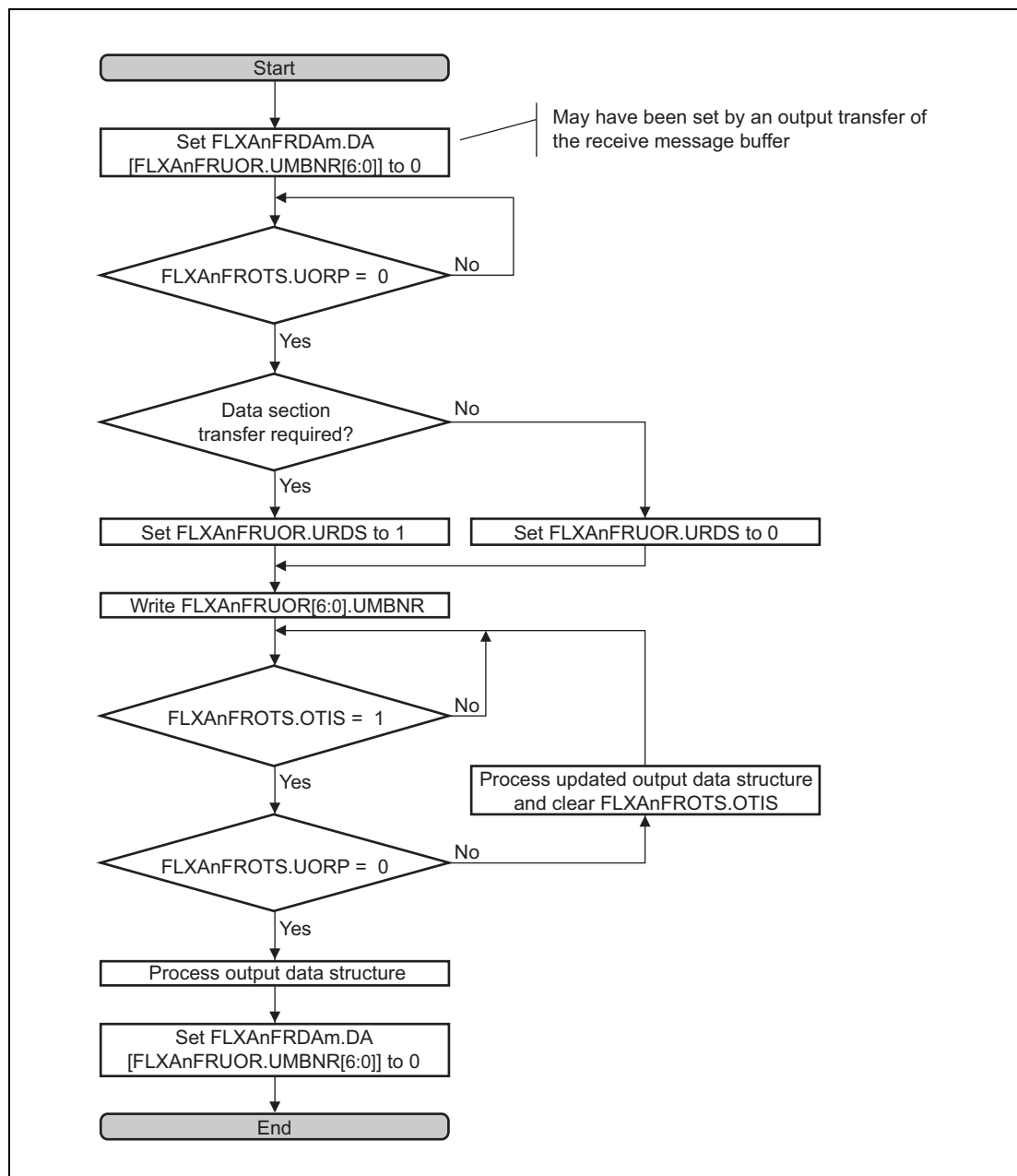


Figure 18.31 User Output Transfer Request Processing Flow

Note that it may be possible that the data structure addressed by a user request is being updated due to a receive message buffer update (which causes bits FLXAnFRDAm.DA[31:0] being set). This set FLXAnFRDAm.DA flags inhibits the user output transfer request. Therefore polling FLXAnFROTS.UORP is not a secure method to identify when the transfer of a requested message buffer has been completed. The bits FLXAnFROTS.OTIS or bits FLXAnFRDAm.DA[31:0] can be used instead. The exact flow depends on the software architecture.

18.3.16.3 Data Structure Transfer Scheduling

Cyclically the different types of transfer requests are checked. In order to guarantee a certain transfer time the different types of transfers have different priorities.

User requested input transfers have highest priority followed by the transfer of data structures committed into the active input transfer queue. No new output transfer will be started as long as there is a pending input transfer request.

The three output transfer request types are checked in a specific order:

(1) All dedicated message buffers in ascending order

When FLXAnFROTC.OTCS is set to 0, set flags FLXAnFRNDATm.NDp are causing a transfer of the message buffer to the output data structure if the destination area is free (bits FLXAnFRDAm.DA[31:0] are 0).

When FLXAnFROTC.OTCS is set to 1, set flags FLXAnFRNDATm.NDp or set flags in the FLXAnFRMBSCm.MBC[31:0] register are causing a transfer of the message buffer to the output data structure if the destination area is free (bits FLXAnFRDAm.DA[31:0] are 0).

(2) FlexRay internal FIFO

When the FlexRay internal FIFO is not empty and there is a free destination area, one FIFO message is transferred into the output data structure specified by the FIFO pointer table.

(3) User output request

If there is a pending user output transfer request, one message buffer is transferred into the corresponding output data structure.

The check sequence is suspended when an input transfer occurs.

18.3.16.4 Behavior in Case of Data Transfer Access Error

The memory areas accessed by the data transfer function may be protected by a memory protection unit (MPU). When the MPU flags an access to a protected address caused by an input or output transfer, an access error event is generated and the related bit in the FLXAnFRAES register is set.

The ongoing transfer is immediately terminated but succeeding transfers are processed and may generate further access errors. Any following access errors are only flagged in FLXAnFRAES.MAE. The other status flags are not updated.

(1) Access error during input transfer

When an access error occurs during an input transfer:

- The ongoing transfer is immediately terminated. The FlexRay internal message RAM will not be updated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.IAE is set to 1
- The input pointer table index is flagged in FLXAnFRAES.EIDX[7:0]
- In case of an normal input transfer the to the transfer related bits FLXAnFRDAm.DA[31:0] is set to 0
- In case of a user input transfer request FLXAnFRITS.UIRP is set to 0

With the given status information the application is able to identify and correct the faulty data structure. In addition the application needs to clear the input access error flag (FLXAnFRAES.IAE).

(2) Access error during output transfer

When an access error occurs during an output transfer:

- The ongoing transfer is immediately terminated but the update of the data structure may have started.
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.OAE is set to 1
- The output pointer table index is flagged in FLXAnFRAES.EIDX[7:0]
- In case of an normal output transfer the to the transfer related bits FLXAnFRDAm.DA[31:0] remains 0 and no output transfer interrupt is generated
- In case of a user output transfer request FLXAnFROTS.UORP is set to 0

With the given status information the application is able to identify and correct the faulty data structure. The data structure in the Local RAM/Global RAM cannot be treated as valid.

In addition the application needs to clear the output access error flag (FLXAnFRAES.OAE).

The FlexRay module internal transfer of the message buffer is completed before the Local RAM/Global RAM access error is detected. The output transfer will not be re-initiated. To avoid loss of data, the application can perform a user output transfer request of this message buffer to a correct Local RAM/Global RAM location.

(3) Access error during FIFO transfer

When an access error occurs during an FIFO transfer:

- The ongoing transfer is immediately terminated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.FAE is set to 1
- The FIFO pointer table index is flagged in FLXAnFRAES.EIDX[7:0]
- The FIFO index pointer are not changed and hence the FIFO status flags are unchanged

With the given status information the application is able to identify and correct the faulty data structure.

In addition the application needs to clear the FIFO access error flag (FLXAnFRAES.FAE).

The data in the Local RAM/Global RAM cannot be treated as valid and is not released to the application. The message cannot be recovered.

18.3.16.5 Behaviors in Case of RAM Read Errors

The FlexRay internal message RAM has an ECC checking mechanism. In case an uncorrectable RAM read error occurs, the application has to analyze the status in the FLXAnFRMHDS register and react as described in **Section 18.3.16.3, Data Structure Transfer Scheduling**. The input and output transfer handler reacts also on these errors detected in the message RAM when the error is related to an active transfer.

In addition, the TBFA and TBFB have an ECC checking mechanism as well. An uncorrectable RAM read errors does not impact the data transfer functionality but have to be handled as described in **Section 18.3.13.1 (4), Message Buffer Status FLXAnFRMBS (word 3)**.

In all cases, data causing a read error is never transferred to the Local RAM/Global RAM. If there is no recovery available in the application, the message is lost.

(1) Read error during transfer from TBF to MBF

This internal transfer is done for each valid FlexRay message received.

A read error can only occur when reading the header section in the FlexRay Message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

For dedicated receive message buffers, the related flags FRNDATm.ND[31:0] will not get set. Consequently the affected message buffer will not be transferred to the output data structure.

For the FlexRay internal FIFO buffers, flags FRNDATm.ND[31:0] are not set but the FlexRay internal FIFO put index is incremented. Due to this, a transfer procedure from the FlexRay internal FIFO buffer to the output buffer is started. However, if the read error is still present in the header section, updating of the output data structure will not start (see **Section 18.3.16.5 (2), Read error during transfer from MBF to OBF**); thus the data in the Local RAM/Global RAM remains correct.

Note that the correction or any other reconfiguration of FIFO related to message buffers while there are pending FIFO transfers may result in incorrect data in the Local RAM/Global RAM. It is strongly recommended to deactivate the output data transfer before starting the reconfiguration and flush the FlexRay internal FIFO before reactivation of the output data transfer.

(2) Read error during transfer from MBF to OBF

This internal transfer is done for every output data transfer (dedicated reception, FIFO, user requested).

A read error can occur in the header and data section (see read error flags in FLXAnFRMHDS). In both cases the message gets lost. If the error is located in the header section, the message buffer needs to be re-configured. If the error is located in the data section, the error is corrected with the next data section update.

When a read error occurs during the transfer from the message RAM to the output buffer, the output data structure will not be updated and the data available will not be set to 1. The FIFO put index and the FIFO fill level are not changed also.

In case of user output transfer request, FLXAnFROTS.UORP is set to 0 even if there was no update of the output data structure.

(3) Read error during transfer from IBF to MBF

This internal transfer is done for every input data transfer.

A read error can occur only when there is no update of the header section requested (the bit LHS in FLXAnFRWRHS4 is set to 0) due to the reading of the header section from the message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

When a read error occurs during the input data transfer, the actually transferred message in the input queue gets lost.

(4) Message RAM read errors

Read errors when reading the header section are flagged in the FLXAnFRMHDS register.

Depending on the buffer type and set buffer protection, a reconfiguration of the message buffer may not be possible.

The input transfer function cannot be used to reconfigure a locked message buffer using the method described in **Section 18.3.13.4 (3), Temporary Unlocking of Header Section**.

Before reconfiguring a locked buffer, the user should disable the input transfer function and the output transfer function.

Section 19 Single Edge Nibble Transmission (RSENT)

The Single Edge Nibble Transmission encoding scheme (SENT) is intended for use in applications where high resolution sensor data needs to be communicated from a sensor to an Engine Control Unit (ECU). It is intended as a replacement for the lower resolution methods of 10 bit A/D's and PWM and as a simpler low cost alternative to CAN or LIN. The implementation assumes that sensor is a smart sensor containing a microprocessor or dedicated logic device (ASIC) to create the signal. RSENT is implemented 5 channels at 100 pin-product and 6 channels at 144 pin-product.

19.1 Features of RH850/P1M-E RSENT

19.1.1 Number of Channels

This product is provided with a RSENT with the following number of channels.

Table 19.1 Number of Channels

| Product | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|--------------------|---|---|
| Number of channels | 5 | 6 |
| Name | RSENT0, RSENT1, RSENT3, RSENT4, RSENT5 | RSENT0, RSENT1, RSENT2, RSENT3, RSENT4, RSENT5 |

Table 19.2 Index

| Index | Meaning |
|-------|---|
| n | This section identifies each RSENT channel by "n" (n = 0 to 5). |

19.1.2 Register Base Address

RSENT register base address are represented by an offset from the base address <RSENTn_base>.

The following table shows the base address <RSENTn_base> of each RSENT module.

Table 19.3 Register Base Address

| RSENTn Channel | <RSENTn_base> Address |
|----------------|------------------------|
| RSENT0 | FFE0 5000 _H |
| RSENT1 | FFE0 6000 _H |
| RSENT2 | FFE0 7000 _H |
| RSENT3 | FFE0 8000 _H |
| RSENT4 | FFE0 9000 _H |
| RSENT5 | FFE0 A000 _H |

19.1.3 Clock Supply

The following clock input is supplied for the RSENT module.

Table 19.4 Clock Supply

| Module | Clock | Supply Clock Name |
|--------|--------------|-------------------------------------|
| RSENTn | pclk clkc | High-speed peripheral clock CLK_HSB |

19.1.4 Interrupt Requests

The RSENT module can generate the following interrupt requests.

Table 19.5 Interrupt Requests

| Unit Interrupt Name | Outline | Interrupt | DMA/DTS Trigger |
|---------------------|-------------------------|-----------|-----------------|
| RSENT0 | | | |
| INTSENT0SI | SENT0 status interrupt | 232 | — |
| INTSENT0RI | SENT0 receive interrupt | 233 | 119 |
| RSENT1 | | | |
| INTSENT1SI | SENT1 status interrupt | 234 | — |
| INTSENT1RI | SENT1 receive interrupt | 235 | 120 |
| RSENT2 | | | |
| INTSENT2SI | SENT2 status interrupt | 236 | — |
| INTSENT2RI | SENT2 receive interrupt | 237 | 121 |
| RSENT3 | | | |
| INTSENT3SI | SENT3 status interrupt | 238 | — |
| INTSENT3RI | SENT3 receive interrupt | 239 | 122 |
| RSENT4 | | | |
| INTSENT4SI | SENT4 status interrupt | 240 | — |
| INTSENT4RI | SENT4 receive interrupt | 241 | 123 |
| RSENT5 | | | |
| INTSENT5SI | SENT5 status interrupt | 242 | — |
| INTSENT5RI | SENT5 receive interrupt | 243 | 124 |

19.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

19.1.6 External Input/Output Signals

External input/output signals of RSENT are listed below.

Table 19.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|------------------|-----|----------------------------|----------------------------------|
| RSENT0 | | | |
| sent_rx | I | RSENT data input | SENT0RX |
| sent_spc | O | RSENT SPC extension output | SENT0SPCO |
| RSENT1 | | | |
| sent_rx | I | RSENT data input | SENT1RX |
| sent_spc | O | RSENT SPC extension output | SENT1SPCO |
| RSENT2 | | | |
| sent_rx | I | RSENT data input | SENT2RX |
| sent_spc | O | RSENT SPC extension output | SENT2SPCO |
| RSENT3 | | | |
| sent_rx | I | RSENT data input | SENT3RX |
| sent_spc | O | RSENT SPC extension output | SENT3SPCO |
| RSENT4 | | | |
| sent_rx | I | RSENT data input | SENT4RX |
| sent_spc | O | RSENT SPC extension output | SENT4SPCO |
| RSENT5 | | | |
| sent_rx | I | RSENT data input | SENT5RX |
| sent_spc | O | RSENT SPC extension output | SENT5SPCO |

19.1.7 Combinations of Pins and Ports

Combinations of RSENT pins and ports are listed in the following table.

Table 19.7 Combinations of Pins and Ports

| Function | Pin Name | Port Name | |
|----------|-----------|---------------------------------|-----------------------------------|
| | | Group 1 | Group 2 |
| RSENT0 | SENT0RX | P5_5 | — |
| | SENT0SPCO | P0_8*1 / P5_5 / P5_6 | — |
| RSENT1 | SENT1RX | P5_8*1 / P5_10 | — |
| | SENT1SPCO | P0_7*1 / P5_8*1 / P5_9 / P5_10 | — |
| RSENT2 | SENT2RX | P5_12*1 | — |
| | SENT2SPCO | P0_6*1 / P5_12*1 / P5_13*1 | — |
| RSENT3 | SENT3RX | P5_14 | P5_11*1 |
| | SENT3SPCO | P0_0 / P0_5*1 / P5_14 / P5_15*1 | P0_0 / P0_5*1 / P5_11*1 / P5_15*1 |
| RSENT4 | SENT4RX | P0_1 | — |
| | SENT4SPCO | P0_1 / P0_4*1 / P3_11 | — |
| RSENT5 | SENT5RX | P0_2 / P2_10*1 | — |
| | SENT5SPCO | P0_2 / P0_3*1 | P2_10*1 / P2_11*1 |

Note 1. Available in devices with 144-pin.

19.2 Functions

Overview of Functions

The RSENT interface supports the following standard specification (SAE J2716 version JAN2010) functions:

- Triple speed expansion Tick Time: Clock cycle (1 μ s to 90 μ s)
- Variable data transmission rate
 - 24.7 kbps to 64.9 kbps: 3 clock rate 6 nibble data
 - 74.1 kbps to 194.7 kbps: 1 clock rate 6 nibble data
- Unidirectional communication: Between the sensor and MCU
- Bidirectional communication: Between the sensor and MCU (supported in SPC mode)
- Single edge data transmission: Coded by the temporal distance of two serially-detected rising edges on a data line
- Transmission frame with up to 6 data nibbles + status and communications nibbles
- Data transmission protected with CRC is available.
 - CRC data can be read with the RSENTnSRXD.SCRC bits.
- Calibration phrase in each data frame
- 1-wire interface (sent_rx and sent_spc share a single terminal.)
- Multiple sensors can connect to the RSENT channel that has the standard expansion function. Received data from sensors is detected by software or DMA.
- The timestamp function: Master or slave can be selected for the RSENT module.
- The RSENT circuit consists of the following functions:
 - Data receive part
 - Clock recovery
 - Register group

19.2.1 Block Diagram

The following figure shows a block diagram of the RSENT module.

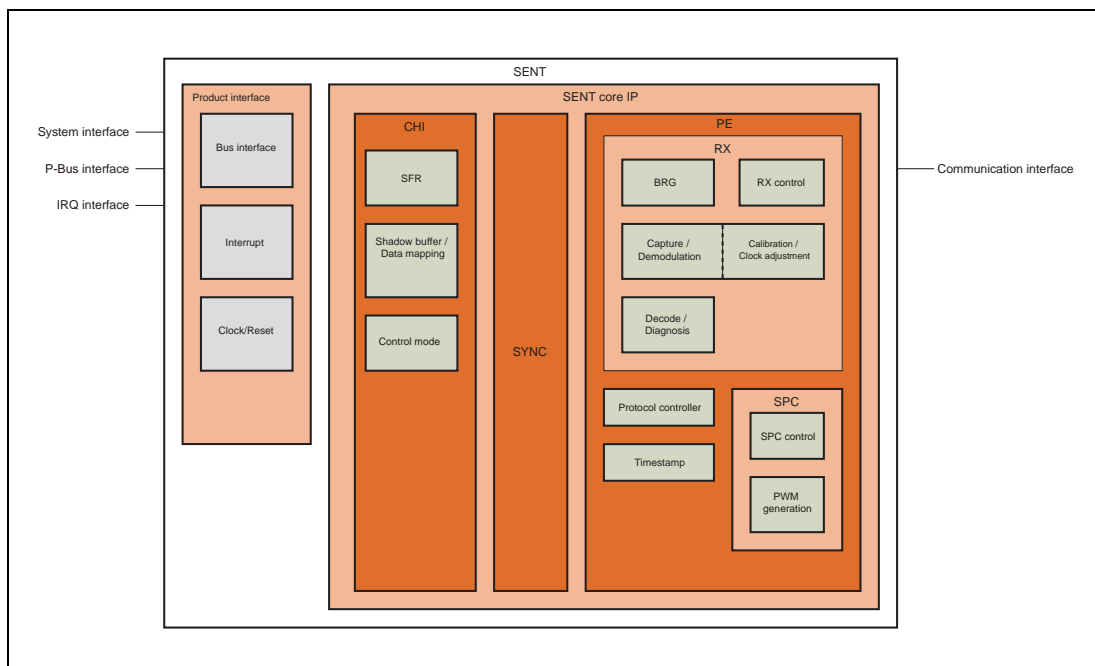


Figure 19.1 Block Diagram of RSENT

19.3 Registers

RSENTn is controlled and operated by the following registers.

For <RSENTn_base>, refer to **Section 19.1.2, Register Base Address**.

Table 19.8 Overview of RSENTn Registers

| Register Name | Abbreviation | Address |
|---|--------------|-----------------------------------|
| RSENT timestamp register | RSENTnTSPC | <RSENTn_base> + 0000 _H |
| RSENT timestamp counter | RSENTnTSC | <RSENTn_base> + 0004 _H |
| RSENT communications configuration register | RSENTnCC | <RSENTn_base> + 0010 _H |
| RSENT baud rate prescaler register | RSENTnBRP | <RSENTn_base> + 0014 _H |
| RSENT interrupt/DMA enable register | RSENTnIDE | <RSENTn_base> + 0018 _H |
| RSENT mode control register | RSENTnMDC | <RSENTn_base> + 001C _H |
| RSENT SPC transmission register | RSENTnSPCT | <RSENTn_base> + 0020 _H |
| RSENT mode status register | RSENTnMST | <RSENTn_base> + 0024 _H |
| RSENT communication status register | RSENTnCS | <RSENTn_base> + 0028 _H |
| RSENT communication status clear register | RSENTnCSC | <RSENTn_base> + 002C _H |
| RSENT slow channel receive timestamp register | RSENTnSRTS | <RSENTn_base> + 0030 _H |
| RSENT slow channel receive data register | RSENTnSRXD | <RSENTn_base> + 0034 _H |
| RSENT calibration pulse length register | RSENTnCPL | <RSENTn_base> + 0038 _H |
| RSENT message length register | RSENTnML | <RSENTn_base> + 003C _H |
| RSENT fast channel receive timestamp register | RSENTnFRTS | <RSENTn_base> + 0040 _H |
| RSENT fast channel receive data register | RSENTnFRXD | <RSENTn_base> + 0044 _H |
| RSENT timestamp mode selection register | RSENTTSEL | <RSENT0_base> + A000 _H |

19.3.1 RSENTnTSPC — RSENT Timestamp Register

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 0000_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----------|-----|-----|-----|-----|-----|-----|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TMS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | TTM[6:0] | | | | | | — | TTPV[6:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 19.9 RSENTnTSPC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 17 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 16 | TMS | Timestamp Mode Selection 0: Master mode 1: Slave mode |
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 14 to 8 | TTM[6:0] | Timestamp Tick Multiplier 000000 _B : 1 000001 _B : 2 000010 _B : 3 : 111111 _B : 128 |
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | TTPV[6:0] | Timestamp Tick Prescaler Value 000000 _B : 1 000001 _B : 2 000010 _B : 3 : 111111 _B : 128 |

RSENTnTSPC.TMS (Timestamp Mode Selection)

When this bit is set to 0, the timestamp counter operates in master mode.

Writing 0000 0000_H to RSENTnTSC by the module set as the master leads to clearing of the timestamps in that module and in the channels set as slaves.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The RSENT operating in slave mode needs to have the same settings in the RSENT that is operated as a master and timestamp counter prescaler.

RSENTnTSPC.TTM (Timestamp Tick Multiplier)

These bits define the multiplication value of the 1- μ s time tick used for the timestamp counter.

For timestamp clock configuration, see **Section 19.5.1, Timestamp**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnTSPC.TTPV (Timestamp Tick Prescaler Value)

These bits define the prescaler value to generate a 1- μ s clock tick.

For timestamp clock configuration, see **Section 19.5.1, Timestamp**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU should configure this value in such a way that, based on the supplied communication clock, a 1- μ s clock tick is generated.

19.3.2 RSENTnTSC — RSENT Timestamp Counter

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 0004_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TS[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 19.10 RSENTnTSC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|-------------------------|
| 31 to 0 | TS[31:0] | Timestamp counter value |

RSENTnTSC.TS (Timestamp)

These bits indicate the current timestamp counter value.

The CPU can only write desired values to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

When the timestamp counter is configured to operate in slave mode (RSENTnTSPC.TMS = 1), writing to this register has no effect when the RSENT module is in either of the OPERATION IDLE or OPERATION ACTIVE modes (the RSENTnMST.OMS bits are either 011_B or 101_B).

The timestamp counter is incremented on every timestamp counter tick (as configured in the RSENTnTSPC.TTPV and RSENTnTSPC.TTM bits) when the RSENT module is in either of the OPERATION IDLE or OPERATION ACTIVE modes (the RSENTnMST.OMS bits are either 011_B or 101_B).

When the timestamp counter is configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU writes 0000 0000_H to these bits and RSENTnTSC.TS is set to 0000 0000_H.

When the slave mode setting is made for the timestamp counter of channel n (RSENTnTSPC.TMS = 1), writing to the timestamp counter of the channel that is the master for channel n leads to the RSENTnTSC.TS bits being set to 0000 0000_H.

For timestamp mode selection, see **Section 19.5.1, Timestamp**.

19.3.3 RSENTnCC — RSENT Communications Configuration Register

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 0010_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|------|-----|------|------|----|----------|------|-----|----------|-----|------|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | SOPC | FCM | SCCD | FCCD | — | SMF[1:0] | PPTC | PPC | NDN[2:0] | | SPCE | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 19.11 RSENTnCC Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 13 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 12 | SOPC | SPC Output Polarity Control 0: SPC pulse active high 1: SPC pulse active low |
| 11 | FCM | Frame Check Method 0: Check against next calibration pulse 1: Check against previous calibration pulse |
| 10 | SCCD | Slow Channel CRC Check 0: Slow channel CRC check enabled 1: Slow channel CRC check disabled |
| 9 | FCCD | Fast Channel CRC Check 0: Fast channel CRC check enabled 1: Fast channel CRC check disabled |
| 8 | — | When read, the value after reset is read. When writing, write the value after reset. |
| 7, 6 | SMF | Serial Message Format 00 _B : No serial message extraction 01 _B : Short serial message format 10 _B : Enhanced serial message format 11 _B : Setting prohibited |
| 5 | PPTC | Pause Pulse Type Configuration 0: Pause pulse for variable message length 1: Pause pulse for fixed message length |
| 4 | PPC | Pause Pulse Configuration 0: Pause pulse absent 1: Pause pulse present |
| 3 to 1 | NDN[2:0] | Number of Data Nibbles 000 _B : 1 data nibble 001 _B : 2 data nibbles 010 _B : 3 data nibbles 011 _B : 4 data nibbles 100 _B : 5 data nibbles 101 _B : 6 data nibbles Other than above: Setting prohibited |

Table 19.11 RSENTnCC Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 0 | SPCE | SPC Mode Enable 0: SPC mode disabled 1: SPC mode enabled |

RSENTnCC.SOPC (SPC Output Polarity Control)

When this bit is set to 0, the SPC pulse is sent as an active high signal. The default output value is low level.

When this bit is set to 1, the SPC pulse is sent as an active low signal. The default output value is high level.

For SPC operation, see also **Section 19.7, SPC Function**.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

NOTE

Any change to this bit from the default value becomes effective on the output value when entering the OPERATION_ACTIVE mode (MST.OMS is 101_B). When entering RESET mode (MST.OMS is 000_B), the output level is set to the default value (low level).

RSENTnCC.FCM (Frame Check Method)

When this bit is set to 0, the current calibration pulse is compared to the next received calibration pulse.

The buffer update mechanism is operating according to the preferred option as described in SAE J2716 2010.

When this bit is set to 1, the current calibration pulse is compared to the previously received calibration pulse.

The buffer update mechanism is operating according to the second option as described in SAE J2716 2010 which should be only used if extra latency to process the second calibration pulse cannot be tolerated.

For buffer update timings, see also **Section 19.6.2.3, Fast Channel Message Reception**.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.SCCD (Slow Channel CRC Check Disable)

When this bit is set to 1, the CRC check for the slow channel is disabled. In this case, messages are stored in the slow channel message reception buffer with the received CRC.

When this bit is set to 1, the RSENTnCS.SCS bit is not set.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.FCCD (Fast Channel CRC Check Disable)

When this bit is set to 1, the CRC check for the fast channel is disabled. In this case, messages are stored in the fast channel message reception buffer with the received CRC.

When this bit is set to 1, the RSENTnCS.FCS bit is not set.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.SMF (Serial Message Format)

These bits define the serial message format expected to be received for automatic extraction.

When these bits are set to 00_B, no serial message is extracted and the communications and status nibble are stored in the RSENTnSRXD register.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU shall set these bits to 00_B when RSENTnCC.SPCE is set to 1 and more than one sensor is connected to the RSENT module.

RSENTnCC.PPTC (Pause Pulse Type Configuration)

This bit defines the pause pulse type.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU should not set this bit to 1 when the RSENTnCC.PPC bit is set to 0.

RSENTnCC.PPC (Pause Pulse Configuration)

This bit defines the presence or absence of the pause pulse.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.NDN (Number of Data Nibbles)

These bits define the number of data nibbles included in an RSENT message.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.SPCE (SPC Mode Enable)

This bit enables the SPC mode.

For details about SPC mode operation, see also **Section 19.7, SPC Function**.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

19.3.4 RSENTnBRP — RSENT Baud Rate Prescaler Register

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 0014_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|-----------|-----|-----|----------|-----|-----|-----|----|----------|-----------|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | TTF[3:0] | | | | — | TTI[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | SCDV[6:0] | | | | | | — | — | — | SCMV[4:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 19.12 RSENTnBRP Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 28 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 27 to 24 | TTF[3:0] | Time Tick Decimal Fraction 0000 _B : 0.0 μs 0001 _B : 0.1 μs 0010 _B : 0.2 μs : 1000 _B : 0.8 μs 1001 _B : 0.9 μs Other than above: Setting prohibited |
| 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 16 | TTI[6:0] | Time Tick Integer 0000000 _B : 1 μs 0000001 _B : 2 μs 0000010 _B : 3 μs : 1011000 _B : 89 μs 1011001 _B : 90 μs Other than above: Setting prohibited |
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 14 to 8 | SCDV[6:0] | Sample Clock Division Value 0000000 _B : 1 0000001 _B : 2 0000010 _B : 3 : 1111110 _B : 127 1111111 _B : 128 |
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 19.12 RSENTnBRP Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 4 to 0 | SCMV[4:0] | Sample Clock Multiplication Value 00000 _B : 1 00001 _B : 2 00010 _B : 3 : 11110 _B : 31 11111 _B : 32 |

RSENTnBRP.TTF (Time Tick Decimal Fraction)

These bits define the decimal part of the tick length in 0.1- μ s granularity.

For tick length configuration, see **Section 19.5.2.2, RX and SPC Tick Settings**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.TTI (Time Tick Integer)

These bits define the integer part of the tick length.

For tick length configuration, see **Section 19.5.2.2, RX and SPC Tick Settings**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.SCDV (Sample Clock Division Value)

These bits define the division value for the sample clock generation logic.

For RSENTnBRP settings, see **Section 19.5.2.1, RX BRP Setting**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.SCMV (Sample Clock Multiplication Value)

These bits define the multiplication value for the sample clock generation logic.

For RSENTnBRP settings, see **Section 19.5.2.1, RX BRP Setting**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

19.3.5 RSENTnIDE — RSENT Interrupt/DMA Enable Register

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 0018_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | SEIE | SMIE | SCIE | NRIE | CVIE | CLIE | FNIE | FEIE | FMIE | FCIE | FRIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 19.13 RSENTnIDE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | SEIE | Slow Channel Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 9 | SMIE | Slow Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 8 | SCIE | Slow Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 7 | NRIE | No Response Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 6 | CVIE | Calibration Pulse Length Variation Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 5 | CLIE | Calibration Pulse Length Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 4 | FNIE | Fast Channel Nibble Count Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 3 | FEIE | Fast Channel Nibble Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 2 | FMIE | Fast Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 1 | FCIE | Fast Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |
| 0 | FRIE | Fast Channel Receive Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled |

RSENTnIDE.SEIE (Slow Channel Encoding Error Interrupt Enable)

This bit enables the generation of the slow channel encoding error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.SMIE (Slow Channel Message Lost Interrupt Enable)

This bit enables the generation of the slow channel message lost interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.SCIE (Slow Channel CRC Error Interrupt Enable)

This bit enables the generation of the slow channel CRC error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.NRIE (No Response Error Interrupt Enable)

This bit enables the generation of the no response error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

The CPU should not set this bit when the SPC mode is disabled (RSENTnCC.SPCE set to 0).

RSENTnIDE.CVIE (Calibration Pulse Length Variation Error Interrupt Enable)

This bit enables the generation of the calibration pulse length variation error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.CLIE (Calibration Pulse Length Error Interrupt Enable)

This bit enables the generation of the calibration pulse length error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FNIE (Fast Channel Nibble Count Error Interrupt Enable)

This bit enables the generation of the fast channel nibble count error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FEIE (Fast Channel Nibble Encoding Error Interrupt Enable)

This bit enables the generation of the fast channel nibble encoding error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FMIE (Fast Channel Message Lost Interrupt Enable)

This bit enables the generation of the fast channel message lost interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FCIE (Fast Channel CRC Error Interrupt Enable)

This bit enables the generation of the fast channel CRC error interrupt.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FRIE (Fast Channel Receive Interrupt Enable)

This bit enables the generation of the fast channel receive interrupt.

The fast channel receive interrupt can be also used to notify a DMA request.

The CPU cannot write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

19.3.6 RSENTnMDC — RSENT Mode Control Register

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 001C_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | OMC[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 19.14 RSENTnMDC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | OMC[2:0] | Operation Mode Control 000 _B : RESET 001 _B : CONFIGURATION 011 _B : OPERATION IDLE 101 _B : OPERATION ACTIVE Other than above: Setting prohibited |

RSENTnMDC.OMC (Operation Mode Control)

These bits are used to control the operation mode of the RSENT module.

- 000_B: RESET

In RESET mode, the mode can only be changed to CONFIGURATION mode.

- 001_B: CONFIGURATION

In CONFIGURATION mode, the mode can only be changed to RESET mode or OPERATION ACTIVE mode.

- 011_B: OPERATION IDLE

In OPERATION IDLE mode, the mode can be changed to OPERATION ACTIVE mode, CONFIGURATION mode, or RESET mode.

- 101_B: OPERATION ACTIVE

In OPERATION ACTIVE mode, the mode can be changed to OPERATION IDLE mode, CONFIGURATION mode, or RESET mode. However, it is recommended to process to the OPERATION IDLE mode first.

For the recommended methods to change between operation modes, see **Section 19.6.1, Changing Operation Modes.**

- Other than above: Setting prohibited

The CPU should not write any other value than listed above into this register.

The CPU should follow the mode change flows as shown in section **Section 19.6.1, Changing Operation Modes**.

19.3.7 RSENTnSPCT — RSENT SPC Transmission Register

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 0020_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----------|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | TLL[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 19.15 RSENTnSPCT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | TLL[6:0] | Length of the Trigger Low Phase in Ticks 000000 _B : 1 tick 000001 _B : 2 ticks 000010 _B : 3 ticks : 111110 _B : 127 ticks 111111 _B : 128 ticks |

RSENTnSPCT.TLL (Trigger Low Length)

These bits define the length of the SPC trigger pulse.

When the CPU writes to these bits, an SPC trigger pulse with the configured length is sent immediately independently of the current status of the RSENT module.

For details about SPC communication, see **Section 19.7, SPC Function**.

The CPU can only write to these bits if the RSENT module is in the OPERATION ACTIVE mode (the RSENTnMST.OMS bits are 101_B) and SPC communication is enabled (RSENTnCC.SPCE is 1).

It is important to note that two consecutive write access might not cause a no response error as the previous request might not have started yet.

After writing to this register, the CPU should wait for at least one SPC trigger tick before writing again to this register.

19.3.8 RSENTnMST — RSENT Mode Status Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 0024_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | OMS[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.16 RSENTnMST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. |
| 2 to 0 | OMS[2:0] | Operation Mode 000 _B : RESET 001 _B : CONFIGURATION 011 _B : OPERATION IDLE 101 _B : OPERATION ACTIVE Other than above: Reserved |

RSENTnMST.OMS (Operation Mode Status)

These bits indicate the current operation mode.

These bits are read only.

These bits are updated after a mode change request is made in the RSENTnMDC.OMC register.

- 000_B: RESET mode

When in RESET mode, all registers are set to their reset values and write access to all registers except the RSENTnMDC register is disabled.

When in RESET mode, RSENT communication is disabled.

- 001_B: CONFIGURATION mode

When in CONFIGURATION mode, write access to the timestamp registers (RSENTnTSPC and RSENTnTSC register), configuration registers (RSENTnCC and RSENTnBRP register), RSENTnIDE register, and mode control register (RSENTnMDC.OMC) is enabled.

When in CONFIGURATION mode, RSENT communication is disabled.

When entering CONFIGURATION mode, all status registers and receive buffer registers are set to their reset values.

- 011_B: OPERATION IDLE mode

In OPERATION IDLE mode, no reception or SPC trigger transmission is possible.

When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.

- 101_B: OPERATION ACTIVE mode

In OPERATION ACTIVE mode, reception and SPC trigger transmission are possible.

- Other than above: Reserved

19.3.9 RSENTnCS — RSENT Communication Status Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 0028_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | SES | SMS | SCS | NRS | CVS | CLS | FNS | FES | FMS | FCS | FRS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.17 RSENTnCS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 11 | Reserved | When read, the value after reset is read. |
| 10 | SES | Slow channel Encoding Error Interrupt Detection 0: Not detected 1: Detected |
| 9 | SMS | Slow Channel Message Lost Interrupt Detection 0: Not detected 1: Detected |
| 8 | SCS | Slow Channel CRC Error Interrupt Detection 0: Not detected 1: Detected |
| 7 | NRS | No Response Error Interrupt Detection 0: Not detected 1: Detected |
| 6 | CVS | Calibration Pulse Length Variation Error Interrupt Detection 0: Not detected 1: Detected |
| 5 | CLS | Calibration Pulse Length Error Interrupt Detection 0: Not detected 1: Detected |
| 4 | FNS | Fast Channel Nibble Count Error Interrupt Detection 0: Not detected 1: Detected |
| 3 | FES | Fast Channel Nibble Encoding Error Interrupt Detection 0: Not detected 1: Detected |
| 2 | FMS | Fast Channel Message Lost Interrupt Detection 0: Not detected 1: Detected |
| 1 | FCS | Fast Channel CRC Error Interrupt Detection 0: Not detected 1: Detected |
| 0 | FRS | Fast Channel Receive Interrupt Detection 0: Not detected 1: Detected |

RSENTnCS.SES (Slow Channel Encoding Error Status)

This bit represents the slow channel encoding error status.

This bit is read only.

In the short serial message format (RSENTnCC.SMF = 01_B), this bit is set when the sequence on serial start bit (status and communications nibble bit #3) is different from “1000 0000 0000 0000_B” (a single 1 and 15 0s).

In the enhanced serial message format (RSENTnCC.SMF = 10_B), this bit is set after receiving the serial message start frame (sequence on status and communications nibble bit 3 is 0111 1110_B), and if bit 13 or bit 18 are not received as ‘0’.

When this bit is set in the short serial message format the received communication and status nibble is used to assemble a serial message.

When this bit is set in the enhanced serial message format the RSENT module checks the presence of a new start sequence at the same time and uses the received communication and status nibble to assemble a serial message.

This bit is cleared when writing 1 to RSENTnCSC.SEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.SMS (Slow Channel Message Lost Status)

This bit represents the slow channel message lost status.

This bit is read only.

This bit is set when there is an attempt to update the slow channel message reception buffer, but the previous message has not been read yet.

This bit is cleared when writing 1 to RSENTnCSC.SEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.SCS (Slow Channel CRC Error Status)

This bit represents the slow channel CRC error status.

This bit is read only.

This bit is set when a CRC error is detected on the slow channel and the slow channel CRC detection is enabled (RSENTnCC.SCCD is set to 0).

This bit is cleared when writing 1 to RSENTnCSC.SCC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.NRS (No Response Error Status)

This bit represents the no response error status.

This bit is read only.

This bit is set when

- the CPU writes to the RSENTnSPCT.TLL bits and
- SPC mode enabled (RSENTnCC.SPCE set to 1) and
- no complete response was received from the sensor for the previous SPC trigger.

This bit is cleared when writing 1 to RSENTnCSC.NRC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001 (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.CVS (Calibration Pulse Length Variation Error Status)

This bit represents the calibration pulse length variation error status.

This bit is read only.

When RSENTnCC.PPTC is 0, then this bit is set when two successive calibration pulses differ by more than 1.5625%.

When RSENTnCC.PPTC is 1, this bit is never set. In this mode (pause pulse with fixed message length), the CPU needs to check the variation of the ratio of calibration pulse to message length by reading the RSENTnCPL and RSENTnML registers.

This bit is cleared when writing 1 to RSENTnCSC.CVC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.CLS (Calibration Pulse Length Error Status)

This bit represents the calibration pulse length error status.

This bit is read only.

This bit is set when the measured calibration pulse length is less than 42 clock ticks or more than 70 clock ticks (deviation of 25% from specification length (56 clock ticks)).

This bit is cleared when writing 1 to RSENTnCSC.CLC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FNS (Fast Channel Nibble Count Error Status)

This bit represents the fast channel nibble count error status.

This bit is read only.

This bit is set when there is an unexpected number of falling edges between two calibration pulses.

This bit is cleared when writing 1 to RSENTnCSC.FNC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FES (Fast Channel Nibble Encoding Error Status)

This bit represents the fast channel nibble encoding error status.

This bit is read only.

This bit is set when on the fast channel a measured nibble period is less than 12 clock ticks or more than 27 clock ticks.

This bit is cleared when writing 1 to RSENTnCSC.FEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FMS (Fast Channel Message Lost Status)

This bit represents the fast channel message lost status.

This bit is read only.

This bit is set when the fast channel message reception buffer is updated, but the previous messages have not been read yet.

This bit is cleared when writing 1 to RSENTnCSC.FMC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FCS (Fast Channel CRC Error Status)

This bit represents the fast channel CRC error status.

This bit is read only.

This bit is set when a CRC error is detected on the fast channel and the fast channel CRC detection is enabled (RSENTnCC.FCCD is set to 0).

This bit is cleared when writing 1 to RSENTnCSC.FCC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FRS (Fast Channel Receive Status)

This bit represents the fast channel receive status.

This bit is read only.

This bit is set when the fast channel message reception buffer was updated.

This bit is cleared when the CPU reads the RSENTnFRXD.FND bit.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

19.3.10 RSENTnCSC — RSENT Communication Status Clear Register

Access: This register can be read/written in 32-bit units.

Address: <RSENTn_base> + 002C_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | SEC | SMC | SCC | NRC | CVC | CLC | FNC | FEC | FMC | FCC | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 19.18 RSENTnCSC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | SEC | Slow Channel Encoding Error Interrupt Clear 0: — 1: Clear |
| 9 | SMC | Slow Channel Message Lost Interrupt Clear 0: — 1: Clear |
| 8 | SCC | Slow Channel CRC Error Interrupt Clear 0: — 1: Clear |
| 7 | NRC | No Response Error Interrupt Clear 0: — 1: Clear |
| 6 | CVC | Calibration Pulse Length Variation Error Interrupt Clear 0: — 1: Clear |
| 5 | CLC | Calibration Pulse Length Error Interrupt Clear 0: — 1: Clear |
| 4 | FNC | Fast Channel Nibble Count Error Interrupt Clear 0: — 1: Clear |
| 3 | FEC | Fast Channel Nibble Encoding Error Interrupt Clear 0: — 1: Clear |
| 2 | FMC | Fast Channel Message Lost Interrupt Clear 0: — 1: Clear |
| 1 | FCC | Fast Channel CRC Error Interrupt Clear 0: — 1: Clear |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

RSENTnCSC.SEC (Slow Channel Encoding Error Clear)

Writing 1 sets RSENTnCS.SES to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.SMC (Slow Channel Message Lost Clear)

Writing 1 sets RSENTnCS.SMS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.SCC (Slow Channel CRC Error Clear)

Writing 1 sets RSENTnCS.SCS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.NRC (No Response Error Clear)

Writing 1 sets RSENTnCS.NRS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.CVC (Calibration Pulse Length Variation Error Clear)

Writing 1 sets RSENTnCS.CVS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.CLC (Calibration Pulse Length Error Clear)

Writing 1 sets RSENTnCS.CLS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FNC (Fast Channel Nibble Count Error Clear)

Writing 1 sets RSENTnCS.FNS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FEC (Fast Channel Nibble Encoding Error Clear)

Writing 1 sets RSENTnCS.FES to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FMC (Fast Channel Message Lost Clear)

Writing 1 sets RSENTnCS.FMS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FCC (Fast Channel CRC Error Clear)

Writing 1 sets RSENTnCS.FCS to 0.

Writing 0 has no effect.

This bit is always read as 0.

19.3.11 RSENTnSRTS — RSENT Slow Channel Receive Timestamp Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 0030_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | STS[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | STS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.19 RSENTnSRTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--------------------------------|
| 31 to 0 | STS | Slow Channel Receive Timestamp |

RSENTnSRTS.STS (Slow Channel Receive Timestamp)

These bits are read only.

These bits are updated when the slow channel message reception buffer is updated with the timestamp counter value of the last frame provided to the slow channel message.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

19.3.12 RSENTnSRXD — RSENT Slow Channel Receive Data Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 0034_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|-----------|----|----|----|----|----|----|----|------|------------|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | SND | — | SCRC[5:0] | | | | | — | — | — | SMGC | IDD[19:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | IDD[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.20 RSENTnSRXD Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 | SND | Slow Channel New Data 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read. |
| 30 | Reserved | When read, the value after reset is read. |
| 29 to 24 | SCRC[5:0] | Slow Channel CRC Data |
| 23 to 21 | Reserved | When read, the value after reset is read. |
| 20 | SMGC | Slow Channel Configuration Bit Data |
| 19 to 0 | IDD[19:0] | Slow Channel Data / ID Information |

RSENTnSRXD.SND (Slow Channel New Data)

This bit indicates that the slow channel message reception buffer is holding data that has not been read.

This bit is read only.

This bit is set when the slow channel message reception buffer is updated.

This bit is cleared automatically whenever it is read.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.SCRC (Slow Channel CRC)

These bits are representing the slow channel CRC data.

These bits are read only.

These bits are updated when the slow channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.SMGC (Slow Channel Configuration Bit)

This bit represents the slow channel configuration bit data.

This bit is read only.

This bit is updated when the slow channel message reception buffer is updated.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.IDD (ID/ Data)

These bits are representing the slow channel data and ID information.

The alignment within this register depends on the message format. For details, see **Section 19.6.2.5, Slow Channel Message Reception**.

These bits are read only.

These bits are updated when the slow channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

19.3.13 RSENTnCPL — RSENT Calibration Pulse Length Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 0038_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CPLV [16] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CPLV[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.21 RSENTnCPL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 17 | Reserved | When read, the value after reset is read. |
| 16 to 0 | CPLV[16:0] | Calibration pulse length value of received message |

RSENTnCPL.CPLV (Calibration Pulse Length Value)

These bits are used by the CPU to calculate the ratio of two consecutive calibration pulses or the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.

These bits are read only.

Updating of the fast channel message reception buffer is storage of the value counted over the calibration pulse length (one tick time × the number of ticks between calibration pulses) based on the sample clock ($f_{\text{SAMPLE}} = 16 \text{ MHz}$).

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

19.3.14 RSENTnML — RSENT Message Length Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 003C_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | MLV[20:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MLV[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.22 RSENTnML Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. |
| 20 to 0 | MLV[20:0] | Message length of received message |

RSENTnML.MLV (Message Length Value)

These bits are used by the CPU to calculate the ratio of the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.

These bits are read only.

Updating of the fast channel message reception buffer is storage of the value counted over the message length (one tick time × the number of ticks taken to receive the overall message) based on the sample clock ($f_{\text{SAMPLE}} = 16 \text{ MHz}$).

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

19.3.15 RSENTnFRTS — RSENT Fast Channel Receive Timestamp Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 0040_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | FTS[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | FTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.23 RSENTnFRTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--------------------------------|
| 31 to 0 | FTS[31:0] | Fast Channel Receive Timestamp |

RSENTnFRTS.FTS (Fast Channel Receive Timestamp)

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

19.3.16 RSENTnFRXD — RSENT Fast Channel Receive Data Register

Access: This register is read-only in 32-bit units.

Address: <RSENTn_base> + 0044_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|-----|-----------|----|-----------|----|----|-----------|----|----|----|----|----|----|----|----|
| | SNDM | FND | FCCN[1:0] | | FCRC[3:0] | | | ND[23:16] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ND[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 19.24 RSENTnFRXD Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 | SNDM | Slow Channel New Data Mirror 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read. |
| 30 | FND | Fast Channel New Data 0: Fast channel frame data is not updated since last read. 1: Fast channel frame data is updated since last read. |
| 29, 28 | FCCN[1:0] | Fast Channel Status and Communications Nibble[1:0] |
| 27 to 24 | FCRC[3:0] | Fast Channel CRC Data |
| 23 to 0 | ND[23:0] | Fast Channel Nibble Data |

RSENTnFRXD.SNDM (Slow Channel New Data Mirror)

This bit indicates that the slow channel message reception buffer is holding data that has not been read.

This bit is read only.

This bit is set when the slow channel message reception buffer is updated.

This bit is cleared automatically whenever the slow channel new data bit (RSENTnSRXD.SND) is read.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FND (Fast Channel New Data)

This bit indicates that the fast channel message reception buffer is holding data that has not been read.

This bit is read only.

This bit is set when the fast channel message reception buffer is updated.

This bit is cleared automatically whenever it is read.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FCCN (Fast Channel Status and Communications Nibble)

These bits are representing the fast channel status and communications nibble bits [1:0].

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FCRC (Fast Channel CRC)

These bits are representing the fast channel CRC data.

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.ND (Fast Channel Nibble Data)

These bits are representing the fast channel nibble data.

The alignment of the nibble data depends on nibble data count (RSENTnCC.NDN). For details, see **Section 19.6.2.3, Fast Channel Message Reception**.

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

19.3.17 RSENTTSSSEL — RSENT Timestamp Mode Selection Register

This register controls the master channel of the RSENT timestamp.

Set this register when the RSENT module is stopped ($RSENTnMST.OMS = 000_B$).

Timestamp in each RSENT channel can be synchronized with other timestamp in other channel. A timestamp clear signal in each channel can clear other timestamp.

Access: This register can be read/written in 32-bit units.

Address: <RSENT0_base> +A000_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|------------|------------|------------|----|------------|------------|------------|----|------------|------------|------------|----|------------|------------|------------|
| | — | — | — | — | — | — | — | — | — | MSEL 52 | MSEL 51 | MSEL 50 | — | MSEL 42 | MSEL 41 | MSEL 40 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | MSEL 32 | MSEL 31 | MSEL 30 | — | MSEL 22 | MSEL 21 | MSEL 20 | — | MSEL 12 | MSEL 11 | MSEL 10 | — | MSEL 02 | MSEL 01 | MSEL 00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Table 19.25 RSENTTSSSEL Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 20 | MSEL5[2:0] | RSENT5 Master Selection (when the RSENTnTSPC.TMS bit of RSENT5 = 1) 000 _B : No timestamp master 001 _B : RSENT0 is the timestamp master of RSENT5. 010 _B : RSENT1 is the timestamp master of RSENT5. 011 _B : RSENT2 is the timestamp master of RSENT5. 100 _B : RSENT3 is the timestamp master of RSENT5. 101 _B : RSENT4 is the timestamp master of RSENT5. 110 _B : No timestamp master Other than above: Setting prohibited |
| 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 to 16 | MSEL4[2:0] | RSENT4 Master Selection (when the RSENTnTSPC.TMS bit of RSENT4 = 1) 000 _B : No timestamp master 001 _B : RSENT0 is the timestamp master of RSENT4. 010 _B : RSENT1 is the timestamp master of RSENT4. 011 _B : RSENT2 is the timestamp master of RSENT4. 100 _B : RSENT3 is the timestamp master of RSENT4. 101 _B : No timestamp master 101 _B : RSENT5 is the timestamp master of RSENT4. Other than above: Setting prohibited |
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 19.25 RSENTSSEL Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 14 to 12 | MSSEL3[2:0] | RSENT3 Master Selection (when the RSENTnTSPC.TMS bit of RSENT3 = 1) 000 _B : No timestamp master 001 _B : RSENT0 is the timestamp master of RSENT3. 010 _B : RSENT1 is the timestamp master of RSENT3. 011 _B : RSENT2 is the timestamp master of RSENT3. 100 _B : No timestamp master 101 _B : RSENT4 is the timestamp master of RSENT3. 101 _B : RSENT5 is the timestamp master of RSENT3. Other than above: Setting prohibited |
| 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 to 8 | MSSEL2[2:0] | RSENT2 Master Selection (when the RSENTnTSPC.TMS bit of RSENT2 = 1) 000 _B : No timestamp master 001 _B : RSENT0 is the timestamp master of RSENT2. 010 _B : RSENT1 is the timestamp master of RSENT2. 011 _B : No timestamp master 100 _B : RSENT3 is the timestamp master of RSENT2. 101 _B : RSENT4 is the timestamp master of RSENT2. 110 _B : RSENT5 is the timestamp master of RSENT2. Other than above: Setting prohibited |
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 4 | MSSEL1[2:0] | RSENT1 Master Selection (when the RSENTnTSPC.TMS bit of RSENT1 = 1) 000 _B : No timestamp master 001 _B : RSENT0 is the timestamp master of RSENT1. 010 _B : No timestamp master 011 _B : RSENT2 is the timestamp master of RSENT1. 100 _B : RSENT3 is the timestamp master of RSENT1. 101 _B : RSENT4 is the timestamp master of RSENT1. 110 _B : RSENT5 is the timestamp master of RSENT1. Other than above: Setting prohibited |
| 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | MSSEL0[2:0] | RSENT0 Master Selection (when the RSENTnTSPC.TMS bit of RSENT0 = 1) 000 _B : No timestamp master 001 _B : No timestamp master 010 _B : RSENT1 is the timestamp master of RSENT0. 011 _B : RSENT2 is the timestamp master of RSENT0. 100 _B : RSENT3 is the timestamp master of RSENT0. 101 _B : RSENT4 is the timestamp master of RSENT0. 110 _B : RSENT5 is the timestamp master of RSENT0. Other than above: Setting prohibited |

19.4 Modes of Operation

The RSENT module can be in one of the following modes:

- RESET mode
- CONFIGURATION mode
- OPERATION IDLE mode
- OPERATION ACTIVE mode

CPU should follow the mode change flow as shown **Section 19.6.1, Changing Operation Modes**.

The current operation mode status can be seen in the RSENTnMST.OMS bits.

Figure 19.2 shows the possible transitions between the channel modes:

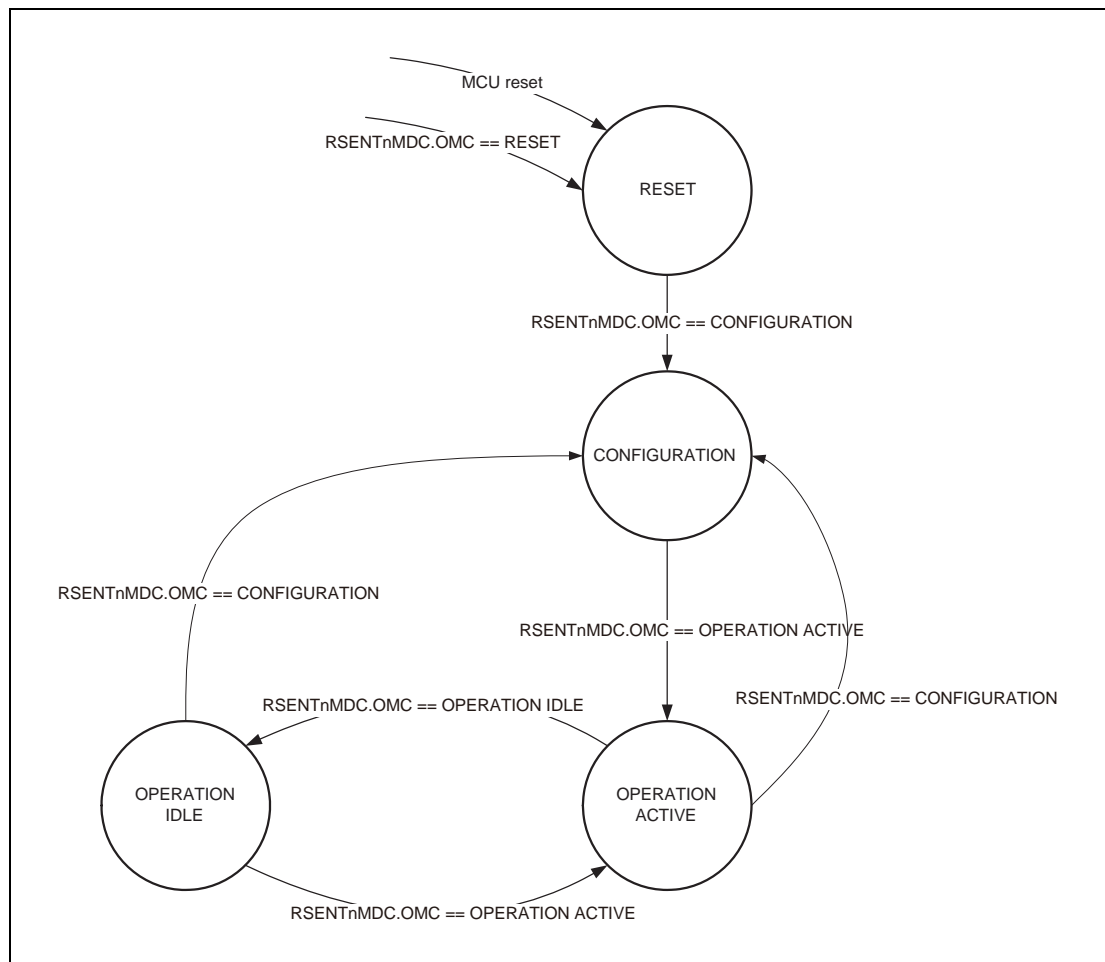


Figure 19.2 Transition between Operation Modes

19.4.1 RESET Mode

This mode is the initial mode that the RSENT module automatically enters after a reset is cleared. Its purpose is to provide a clean reset of the registers in the RSENT module.

The RESET mode is also entered after the RSENTnMDC.OMC bits have been set to 000_B. In this state, all, configuration, control (except RSENTnMDC.OMC bits and bits in the RSENTTSSEL register), and status registers are set to the value after their reset. Any on-going transmission or reception process is stopped immediately and the interface pins of the RSENT module are set to the value after their reset.

Read access to all registers is possible in this state. Write access is limited to the RSENTnMDC register and RSENTTSSEL registers.

19.4.2 CONFIGURATION Mode

The CONFIGURATION mode is entered after the RSENTnMDC.OMC bits have been set to 001_B.

The interface pins of the RSENT module are set to their default values.

Regarding the output polarity setting of RSENTnSPCO pin and the timing that becomes effective, please refer to the explanation of “RSENTnCC.SOPC (SPC Output Polarity Control)” in **Section 19.3.3, RSENTnCC — RSENT Communications Configuration Register**.

In this state, all status registers (RSENTnCS) and the receive buffer registers (RSENTnSRTS, RSENTnSRXD, RSENTnCPL, RSENTnML, RSENTnFRTS, and RSENTnFRXD) are set to the value after their reset.

Read access to all registers is possible in this state.

Write access is limited to both timestamp registers (RSENTnTSPC and RSENTnTSC) and configuration registers (RSENTnCC, RSENTnBRP, RSENTnIDE, RSENTnMDC, and RSENTnCSC).

19.4.3 OPERATION IDLE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 011_B.

In OPERATION IDLE mode, no reception and transmission are done.

When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.

Read access to all registers is possible in this state.

Write access is granted only to RSENTnTSC, RSENTnIDE, RSENTnMDC, and RSENTnCSC.

19.4.4 OPERATION ACTIVE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 101_B.

In OPERATION ACTIVE mode, transmission and reception can take place.

Frame reception and status flagging starts after a valid calibration pulse (including the falling edge at the beginning) was detected.

Read access to all registers is possible in this state.

Write access is granted only to RSENTnTSC, RSENTnIDE, RSENTnMDC, RSENTnSPCT, and RSENTnCSC.

19.4.5 Register Behavior in Operation Modes

Table 19.26 shows the register behavior when the RSENT module transitions to the indicated operation modes. The table also gives an overview about the access restriction in each operation mode.

Table 19.26 Register Behavior in Operation Modes

| Register Name | Symbol | MCU Reset | RESET | | CONFIGURATION | | OPERATION IDLE | | OPERATION ACTIVE | |
|--|------------|------------------------|------------------------|-----|------------------------|-----|------------------------|-------------------|------------------------|-------------------|
| | | Change | Change | R/W | Change | R/W | Change | R/W | Change | R/W |
| Timestamp prescaler configuration register | RSENTnTSPC | 0000 0000 _H | 0000 0000 _H | R | Unchanged | R/W | Unchanged | R | Unchanged | R |
| Timestamp counter register | RSENTnTSC | 0000 0000 _H | 0000 0000 _H | R | Unchanged | R/W | Unchanged | R/W _{*1} | Unchanged | R/W _{*1} |
| Communications configuration register | RSENTnCC | 0000 0000 _H | 0000 0000 _H | R | Unchanged | R/W | Unchanged | R | Unchanged | R |
| Baud rate prescaler register | RSENTnBRP | 0000 0000 _H | 0000 0000 _H | R | Unchanged | R/W | Unchanged | R | Unchanged | R |
| Interrupt/DMA enable register | RSENTnIDE | 0000 0000 _H | 0000 0000 _H | R | Unchanged | R/W | Unchanged | R/W | Unchanged | R/W |
| Mode control register | RSENTnMDC | 0000 0000 _H | 0000 0000 _H | R/W | Unchanged | R/W | Unchanged | R/W | Unchanged | R/W |
| SPC transmission register | RSENTnSPCT | 0000 0000 _H | 0000 0000 _H | R | Unchanged | R | Unchanged | R | Unchanged | R/W |
| Mode status register | RSENTnMST | 0000 0000 _H | 0000 0000 _H | R | 0000 0001 _H | R | 0000 0003 _H | R | 0000 0005 _H | R |
| Communication status register | RSENTnCS | 0000 0000 _H | 0000 0000 _H | R | 0000 0000 _H | R | Unchanged | R | Unchanged | R |
| Communication status clear register | RSENTnCSC | 0000 0000 _H | 0000 0000 _H | R | Unchanged | R/W | Unchanged | R/W | Unchanged | R/W |
| Slow channel receive timestamp register | RSENTnSRTS | 0000 0000 _H | 0000 0000 _H | R | 0000 0000 _H | R | Unchanged | R | Unchanged | R |
| Slow channel receive data register | RSENTnSRXD | 0000 0000 _H | 0000 0000 _H | R | 0000 0000 _H | R | Unchanged | R | Unchanged | R |
| Calibration pulse length register | RSENTnCPL | 0000 0000 _H | 0000 0000 _H | R | 0000 0000 _H | R | Unchanged | R | Unchanged | R |
| Message length register | RSENTnML | 0000 0000 _H | 0000 0000 _H | R | 0000 0000 _H | R | Unchanged | R | Unchanged | R |
| Fast channel receive timestamp register | RSENTnFRTS | 0000 0000 _H | 0000 0000 _H | R | 0000 0000 _H | R | Unchanged | R | Unchanged | R |
| Fast channel receive data register | RSENTnFRXD | 0000 0000 _H | 0000 0000 _H | R | 0000 0000 _H | R | Unchanged | R | Unchanged | R |

Note 1. Means write restriction exists.

19.5 Clock Configuration

19.5.1 Timestamp

19.5.1.1 Timestamp Clock Configuration

RSENT incorporates the timestamp counter.

The user should configure the RSENTnTSPC.TTPV register according to the applied communication frequency. The resolution of the timestamp is 1 μ s. The input frequency is divided by the configured timestamp prescaler value RSENTnTSPC.TTPV.

Depending on the configured tick lengths, the resolution can be decreased by configuring the RSENTnTSPC.TTM bits. The already divided input frequency is divided further by the value of the TSPC.TTM bits.

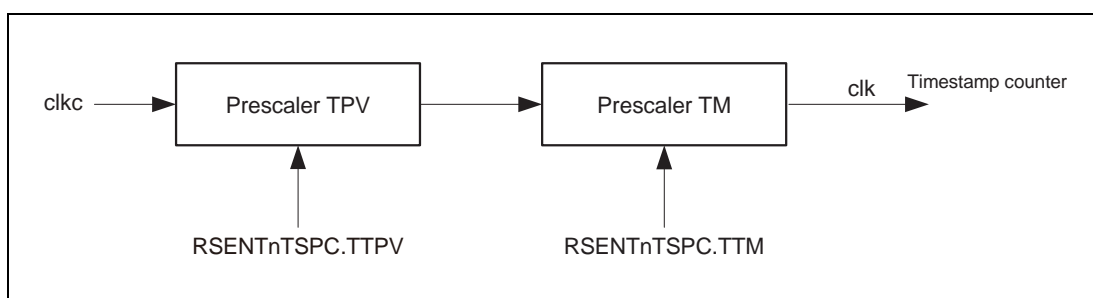


Figure 19.3 Timestamp Counter Clock Generation

19.5.1.2 Timestamp Counter Operation

The timestamp counter value can be initialized to any value by writing to the RSENTnTSC.TS bits only when the RSENT module is in CONFIGURATION mode.

When timestamp counters are configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU can reset the timestamp counter by writing 0000 0000_H to the RSENTnTSC.TS bits when the RSENT module is in OPERATION IDLE or OPERATION ACTIVE mode.

When timestamp counters are configured to operate in slave mode (RSENTnTSPC.TMS = 1), the timestamp counter is cleared when the CPU writes 0000 0000_H to the RSENTnTSC.TS bits of the channel set in the master when the RSENT module is in OPERATION IDLE or OPERATION ACTIVE mode. Make the same settings for the timestamp counter prescalers of channels operating in master mode and slave mode. When timestamp counter synchronization occurs, the internal timestamp counter prescalers are also synchronized.

The current timestamp counter value can be read from the RSENTnTSC.TS bits.

When the RSENT module is in OPERATION ACTIVE mode, each received message is stored with its related timestamp. Timestamp values are taken for fast channel and slow channel data.

The timestamp value is captured when the calibration pulse is detected.

The timestamp value for the fast channel is stored in the RSENTnFRTS.FTS bits.

The timestamp value for the slow channel is stored in the RSENTnSRTS.STS bits. The timestamp value for the slow channel is identical to the timestamp value of the last fast channel message contributing to the slow channel message.

In case timestamp counter synchronization is required, the following flow should be used.

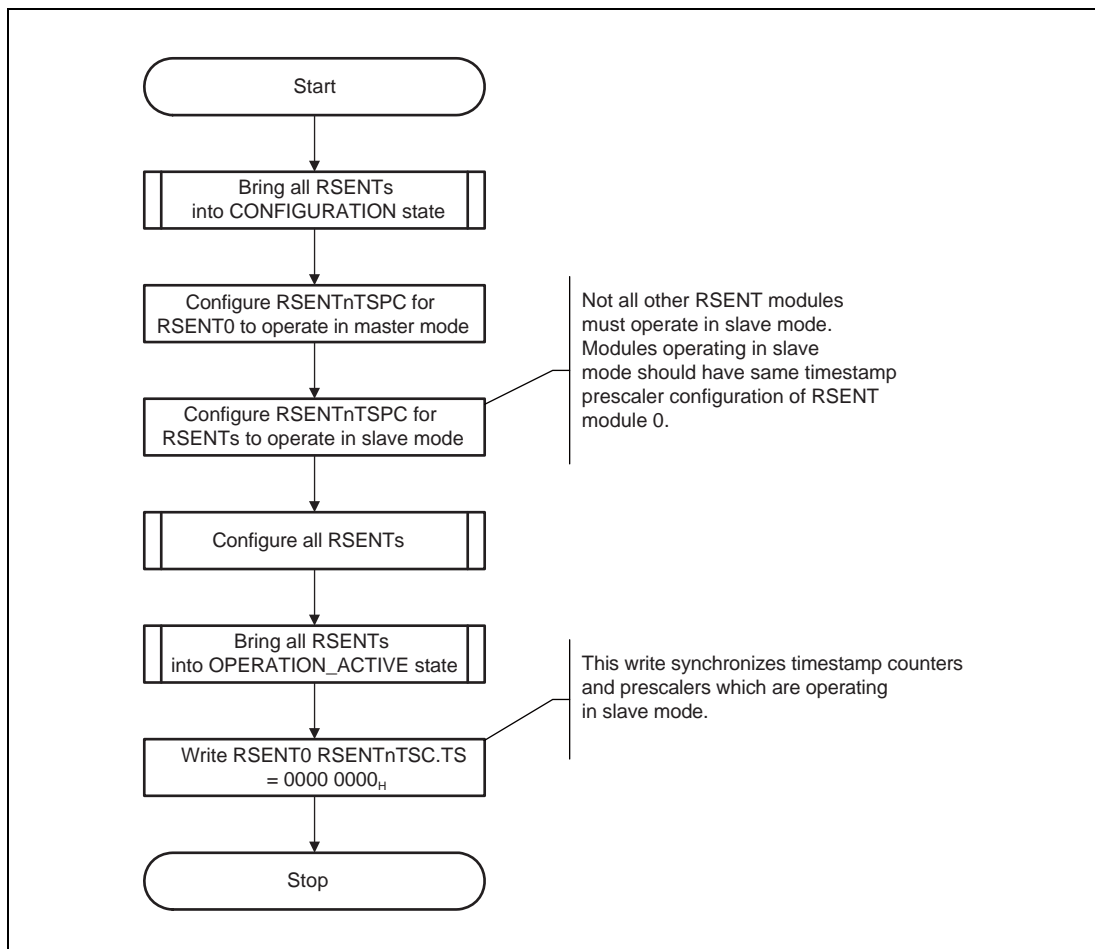


Figure 19.4 Timestamp Counter Synchronization

Synchronization of the timestamp can be done if the state of the master module has changed to OPERARION_ACTIVE or OPERARION_IDLE.

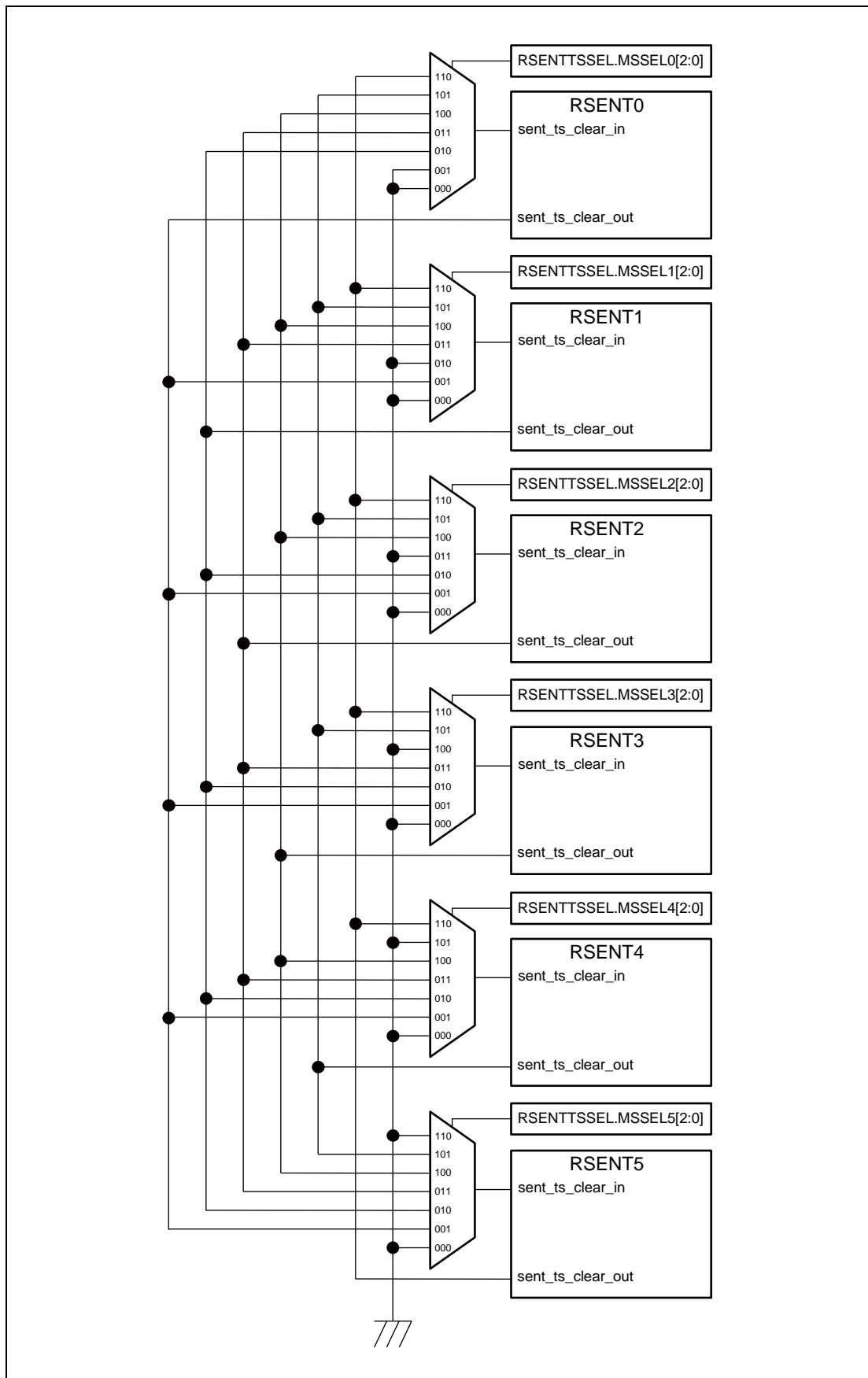


Figure 19.5 Timestamp Signal Connection among Channels

19.5.2 Communication Clock Configuration

19.5.2.1 RX BRP Setting

Use the formula below to obtain the settings for BRP.SCDV and BRP.SCMV in accord with $f_{\text{COMMUNICATION}}$ to be used (the frequency of the clk clock) so that the sample clock frequency is 16 MHz (for example, if the settings are for BRP.SCDV = 4/BRP.SCMV = 6, replace this with BRP.SCDV = 2/BRP.SCMV = 3). The input clock (clk clock) is frequency-divided according to the settings of BRP.SCDV and BRP.SCM to generate the sample clock.

Set $f_{\text{COMMUNICATION}}$ (the frequency of the clk clock) so that it falls within the range from 32 to 100 MHz (or at precisely 16 MHz). Use the formula below to obtain the values for BRP.SCDV and BRP.SCM such that the sample clock frequency becomes 16 MHz.

$$f_{\text{SAMPLE}} = 16 \text{ MHz} = f_{\text{COMMUNICATION}} \times \frac{\text{Sample Clock Multiplication Value}(\text{BRP.SCMV} + 1)}{\text{Sample Clock Division Value} (\text{BRP.SCDV} + 1)}$$

Where Sample Clock Multiplication Value = 1 (BRP.SCMV = 00000_B),

Sample Clock Division Value = 5 (BRP.SCDV = 0000100_B),

$f_{\text{COMMUNICATION}} = 80 \text{ MHz}$

$$f_{\text{SAMPLE}} = 80 \times 1/5 = 16 \text{ MHz}$$

19.5.2.2 RX and SPC Tick Settings

The used tick length in RX and SPC function can be configured with the RSENTnBRP.TTI and RSENTnBRP.TTF bits. Tick lengths from 1.0 μs to 90.0 μs with a resolution of 0.1 μs can be configured.

The RSENTnBRP.TTI holds the integer part of the tick length and the RSENTnBRP.TTF bits hold the fractional part of the tick length. The tick length is then calculated by:

$$T_{\text{TICK}} = T_{\text{BRP.TTI}} + T_{\text{BRP.TTF}}$$

Where BRP.TTI = 0000000_B, BRP.TTF = 0011_B

$$T_{\text{TICK}} = 1 + 0.3 = 1.3 \mu\text{s}$$

19.6 RSENT Operation

19.6.1 Changing Operation Modes

Once initialization has been completed in CONFIGURATION mode, operation can be enabled by entering OPERATION ACTIVE mode. This is done by setting the RSENTnMDC.OMC bits to OPERATION ACTIVE and waiting for the RSENTnMST.OMS to transition to OPERATION ACTIVE.

Once in OPERATION ACTIVE mode the RSENT module begins to receive messages or SPC communication can be started depending on the configuration.

Figure 19.6 shows the communication enabled flow assuming that the RSENT module is in RESET mode:

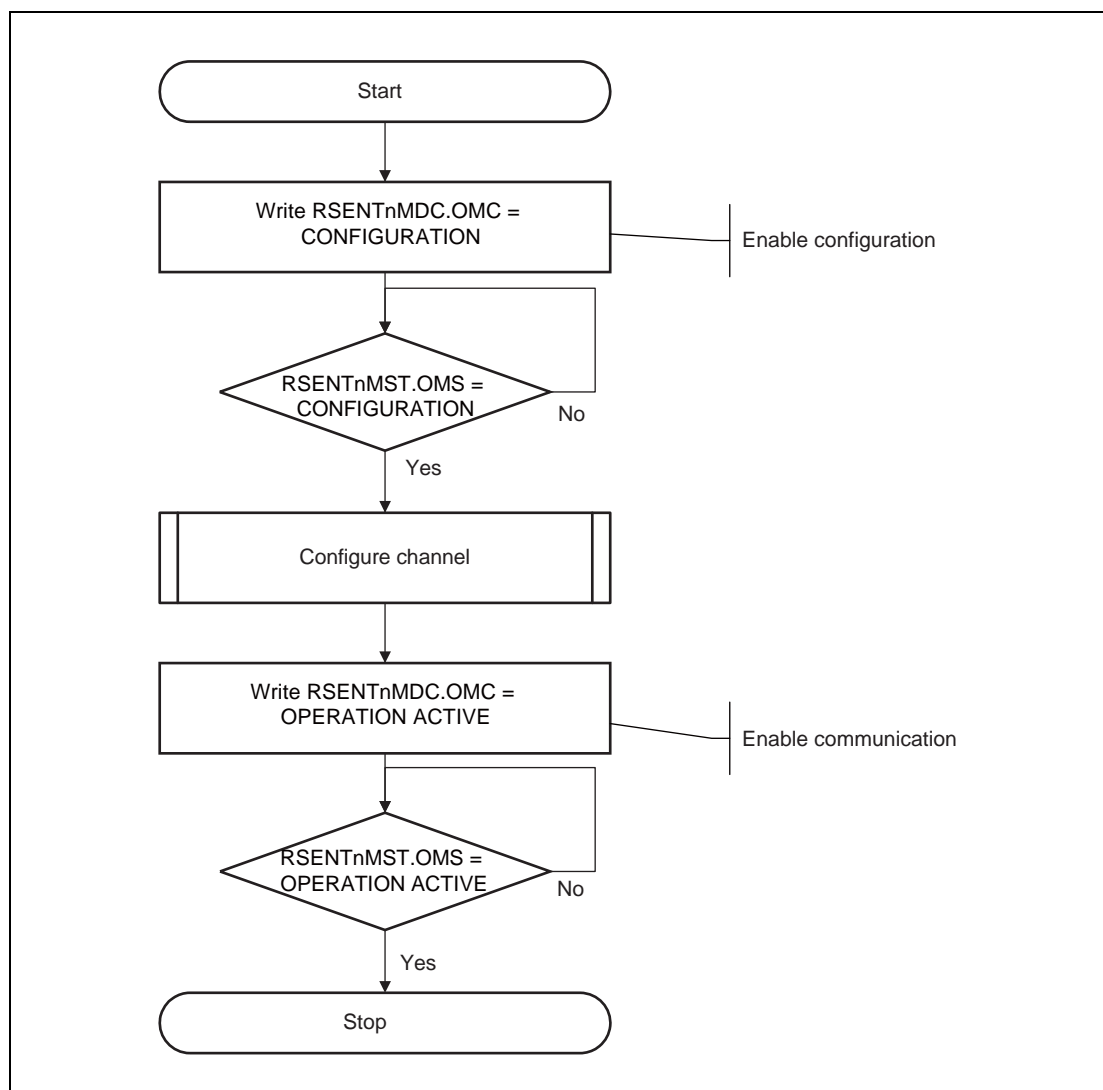


Figure 19.6 Communication Enable Flow

To leave OPERATION ACTIVE mode, communication should be disabled first by entering OPERATION IDLE mode. This is done by setting the RSENTnMDC.OMC bits to OPERATION IDLE and waiting for the RSENTnMST.OMS bits to transition to OPERATION IDLE.

However, when the SPC mode is enabled ($CC.SPCE = 1$) and the SPC trigger transmission has not been requested after the previous SPC communication has been completed (e.g. successful reception for the previous SPC trigger transmission), the RSENT module can directly enter the CONFIGURATION mode.

The transition between OPERATION ACTIVE and OPERATION IDLE depends on the setting of the $RSENTnCC.SPCE$ bit.

(1) $RSENTnCC.SPCE = 0$

In case a reception is currently ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the receive buffer was updated or error was flagged (see **Section 19.6.2.3, Fast Channel Message Reception**).

In case no reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place immediately.

(2) $RSENTnCC.SPCE = 1$

In case a reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

In case a no response error is flagged, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place at the same time as the error flagging.

The mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the sequence of making a SPC trigger and receiving the response has been completed. This means when a response was already received, the transition takes place immediately. When the response is still pending, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

CONFIGURATION mode can be entered by writing CONFIGURATION to the $RSENTnMDC.OMC$ bits and waiting for the $RSENTnMST.OMS$ to transition to CONFIGURATION.

Once CONFIGURATION mode is entered, the remaining status and message information stored in the RSENT module is lost since status and message information is cleared in CONFIGURATION mode.

Figure 19.7 shows the communication disable flow assuming that the RSENT module is in OPERATION ACTIVE mode.

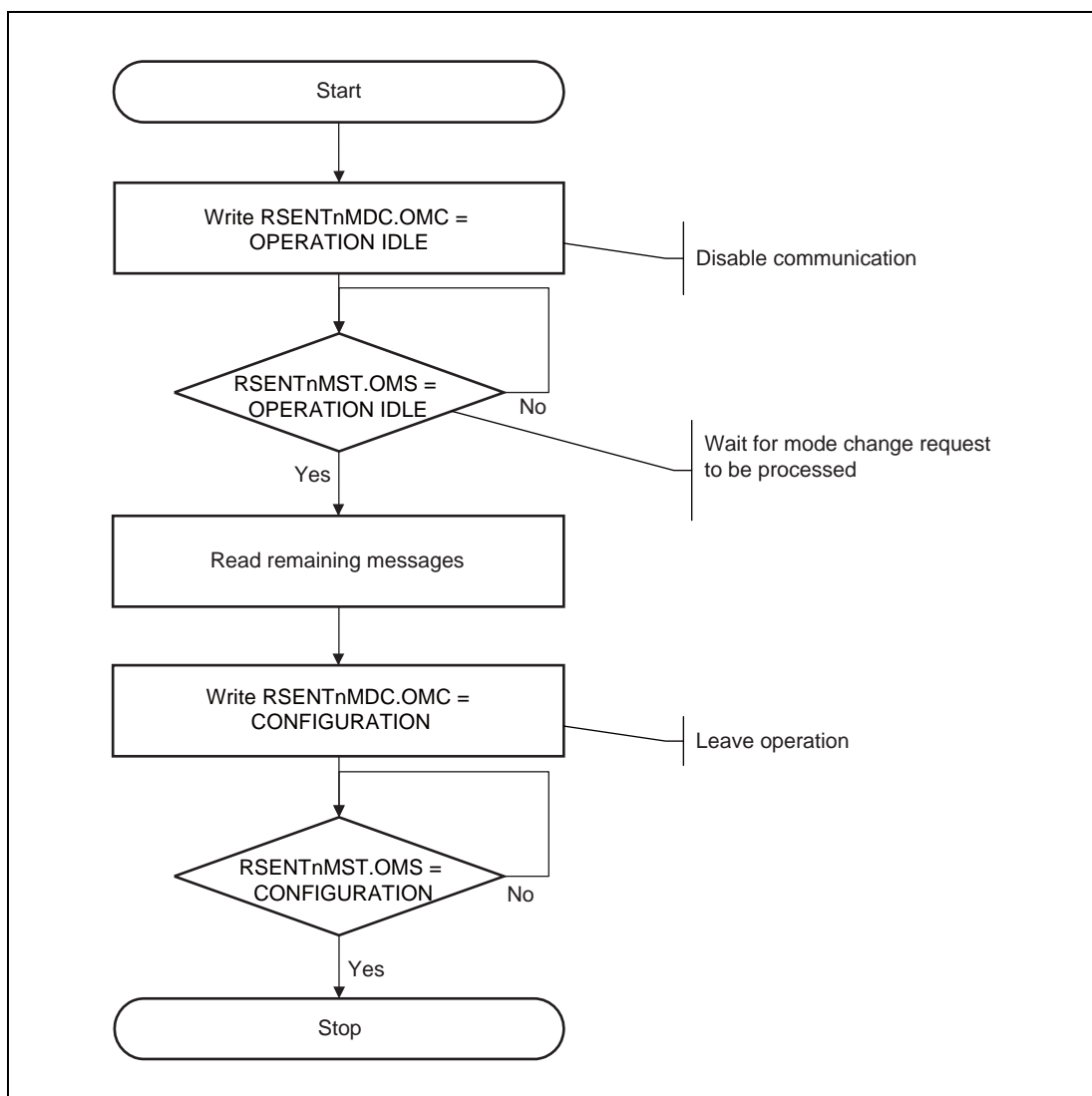


Figure 19.7 Communication Stop Flow

19.6.2 Message Reception

RSENT message reception is composed of the calibration pulse reception followed by the data nibble pulse reception.

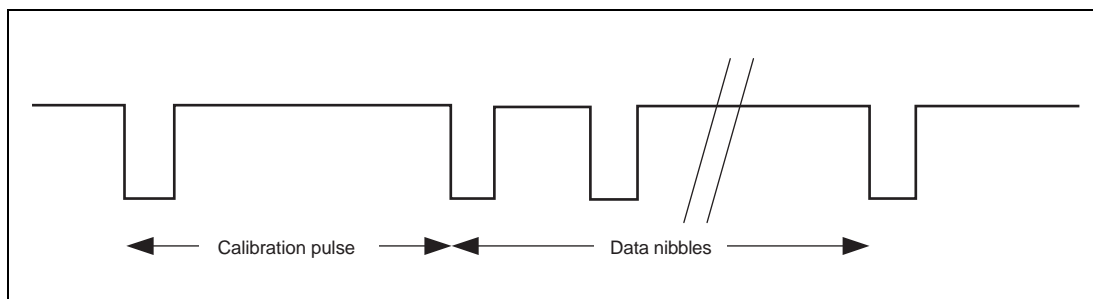


Figure 19.8 RSENT Received Message Structure

19.6.2.1 Calibration Pulse Reception

Within the calibration pulse reception phase the internally generated clock tick is adjusted to the transmit clock speed.

In addition, the calibration pulse is used to end the previous message and perform message diagnostics. The RSENT module supports automatic calibration pulse length diagnostics in variable message length modes (RSENTnCC.PPTC = 0). In case the calibration pulse ratio check fails, the calibration pulse length variation error flag (RSENTnCS.CVS) is set to 1.

19.6.2.2 Data Nibble Reception

The receive function of the RSENT module is a straightforward capture and compare function. The RSENT module receives sensor information encoded by the temporal distance of two consecutive falling edges on the data line. The temporal distance (the number of clock ticks) is captured and compared against a set of values to determine the actual nibble value. The data encoding is illustrated in **Table 19.27** below.

Table 19.27 Data Nibble Encoding

| Nibble Period (Number of Clock Ticks) | Nibble Value (Binary) |
|---------------------------------------|-----------------------|
| 12 | 0000 _B |
| 13 | 0001 _B |
| 14 | 0010 _B |
| 15 | 0011 _B |
| 16 | 0100 _B |
| 17 | 0101 _B |
| 18 | 0110 _B |
| 19 | 0111 _B |
| 20 | 1000 _B |
| 21 | 1001 _B |
| 22 | 1010 _B |
| 23 | 1011 _B |
| 24 | 1100 _B |
| 25 | 1101 _B |
| 26 | 1110 _B |
| 27 | 1111 _B |

The received data nibbles are composed into an RSENT message which is then stored in the fast channel message reception buffer.

Any other received nibble period during the reception of data nibbles will cause a fast channel nibble encoding error.

19.6.2.3 Fast Channel Message Reception

Messages received on the fast message channel are stored in a receive buffer.

A fast channel message reception buffer is composed of the calibration pulse length register (RSENTnCPL), the message length register (RSENTnML), the fast channel receive timestamp register (RSENTnFRTS), and the fast channel receive data register (RSENTnFRXD).

These registers are arranged on successive addresses for a transfer of the register content into memory using DMA.

The RSENT module is equipped with a double receive buffer structure that allows the storage of two complete RSENT messages including the related timestamp and message length information. Message decoding and assembling are done in a separate register stage.

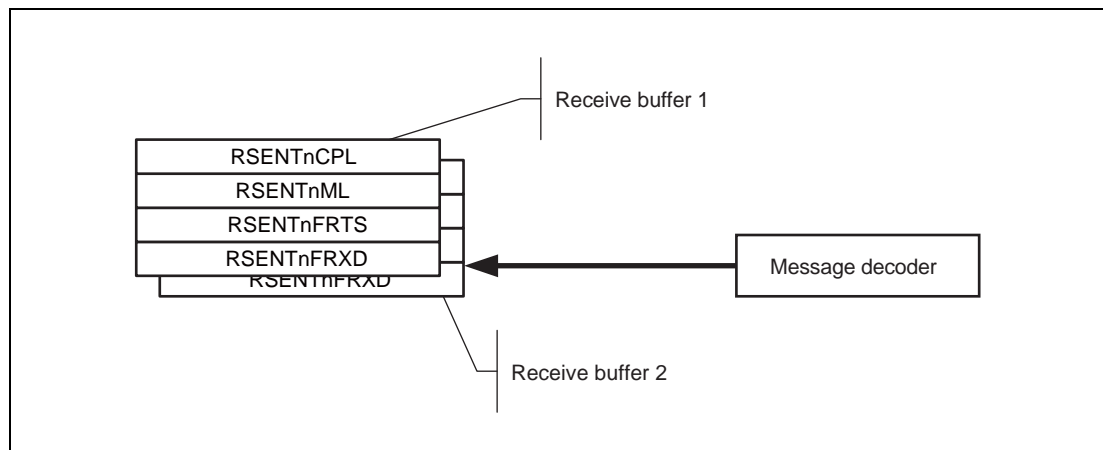


Figure 19.9 Fast Channel Message Reception Buffer

The first received message is placed into the message buffer that can be accessed by the CPU. This buffer (except the RSENTnFRXD.SNDM bit) is not updated any more until the RSENTnFRXD.FND bit was read.

When a new message is placed into a receive buffer, the RSENTnFRXD.FND bit is set. At the same time, the RSENTnCS.FRS bit is set and, if enabled, a receive interrupt request is generated.

When receive buffer 1 is holding an unprocessed message (the RSENTnFRXD.FND bit is 1), any further incoming message is placed in receive buffer 2. Receive buffer 2 is updated with any further incoming messages. In case an unprocessed message in receive buffer 2 is overwritten, the RSENTnCS.FMS bit is set to 1.

When the CPU reads the RSENTnFRXD.FND bit and there is valid data in buffer 2, the data previously located in receive buffer 2 becomes available in the receive buffer and is accessible by the CPU. If enabled, a new interrupt request for fast channel data is generated and RSENTnCS.FRS is set.

When the RSENTnFRXD.FND/ RSENTnCS.FRS bit is not set, the data in the receive buffer is not defined and the CPU should not access the receive buffer.

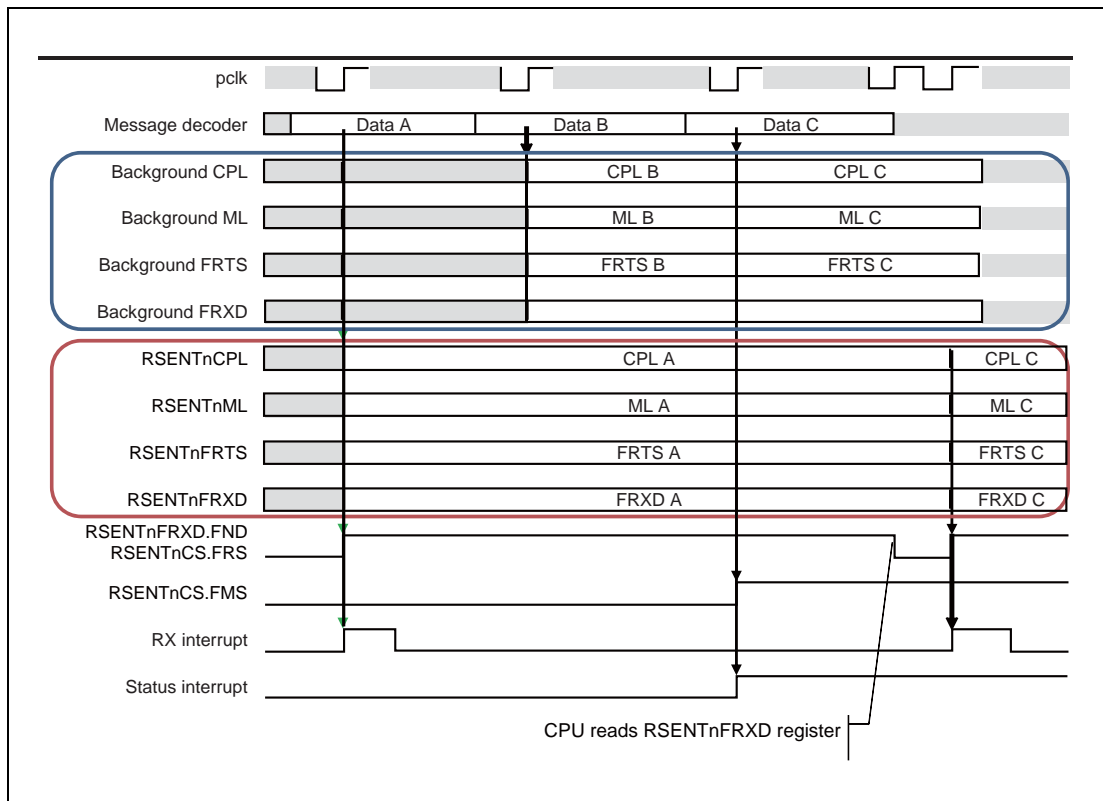


Figure 19.10 Fast Channel Message Reception Buffer Update Timing

The update timing of the receive buffer depends on the applied configuration as depicted in **Figure 19.11** to **Figure 19.14**.

The RSENTnFRTS register is updated with the current timestamp counter register value when the calibration pulse is detected.

The data alignment in the RSENTnFRXD register depends on the nibble data count (RSENTnCC.NDN).

Table 19.28 Data Nibble Alignment in RSENTnFRXD Register

| RSENTnCC.NDN | 23:20 | 19:16 | 15:12 | 11:8 | 7:4 | 3:0 |
|------------------|-----------|-----------|-----------|-----------|-----------|----------|
| 000 _B | Undefined | Undefined | Undefined | Undefined | Undefined | Nibble 1 |
| 001 _B | Undefined | Undefined | Undefined | Undefined | Nibble 1 | Nibble 2 |
| 010 _B | Undefined | Undefined | Undefined | Nibble 1 | Nibble 2 | Nibble 3 |
| 011 _B | Undefined | Undefined | Nibble 1 | Nibble 2 | Nibble 3 | Nibble 4 |
| 100 _B | Undefined | Nibble 1 | Nibble 2 | Nibble 3 | Nibble 4 | Nibble 5 |
| 101 _B | Nibble 1 | Nibble 2 | Nibble 3 | Nibble 4 | Nibble 5 | Nibble 6 |

(1) SAE operation with variable message length and preferred check method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 0)

In this operation mode, the RSENT module automatically performs the check for successive calibration pulse variation according to the preferred option in the J2716 2010 specification. In this mode, message diagnostics is done after the calibration pulse was received following a message.

If this check is passed, the message reception buffer is updated.

If this check is not passed, the message reception buffer is not updated and RSENTnCS.CVS is set to 1.

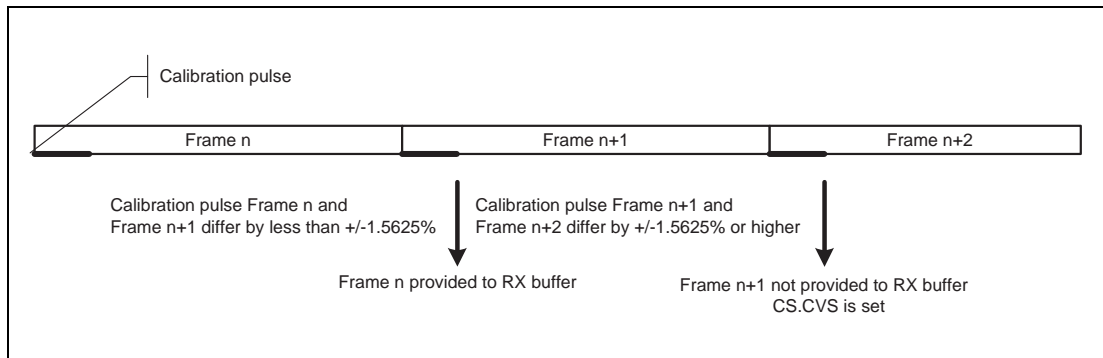


Figure 19.11 Buffer Update in Variable Message Length Mode and Preferred Check Method

(2) SAE operation with variable message length and optional check method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 1)

In this operation mode, the RSENT module automatically performs the check for successive calibration pulse variation according to the optional frame check method as described in the J2716 2010 specification. In this mode, the calibration pulse of the current frame is compared to the calibration pulse of the last valid preceding frame.

If this check is passed, the message reception buffer is updated.

If this check is not passed, the message reception buffer is not updated and RSENTnCS.CVS is set to 1.

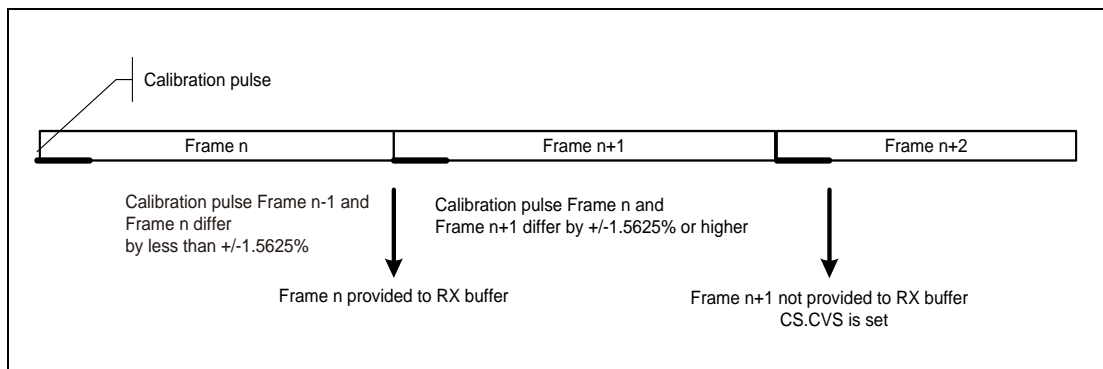


Figure 19.12 Buffer Update in Variable Message Length Mode and Optional Check Method

(3) SAE operation with fixed message length (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 1)

In this mode, the RSENT module does not perform the check for calibration pulse and message length ratio according to the preferred option in the J2716 2010 specification. In this mode, the RSENT module provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples.

The message buffer is updated at the beginning of the following calibration pulse irrespective of the values in the RSENTnCPL and RSENTnML registers. The CPU needs to calculate the ratio and either accept or discard the message.

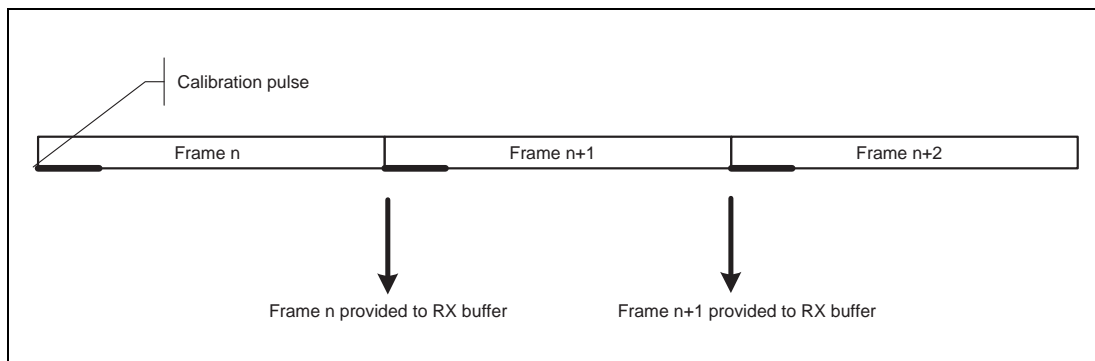


Figure 19.13 Buffer Update in Fixed Message Length Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

(4) SPC operation (RSENTnCC.SPCE = 1)

In this operation mode, sensor data transmission is done following a SPC master trigger pulse. Within SAE SENT communication, the calibration pulse or pause pulse is aborting the previous message. In SPC communication, the sensor is only sending data following a SPC trigger request. An end pulse sent by the sensor is aborting the message. The message buffer is updated at the beginning of the end pulse.

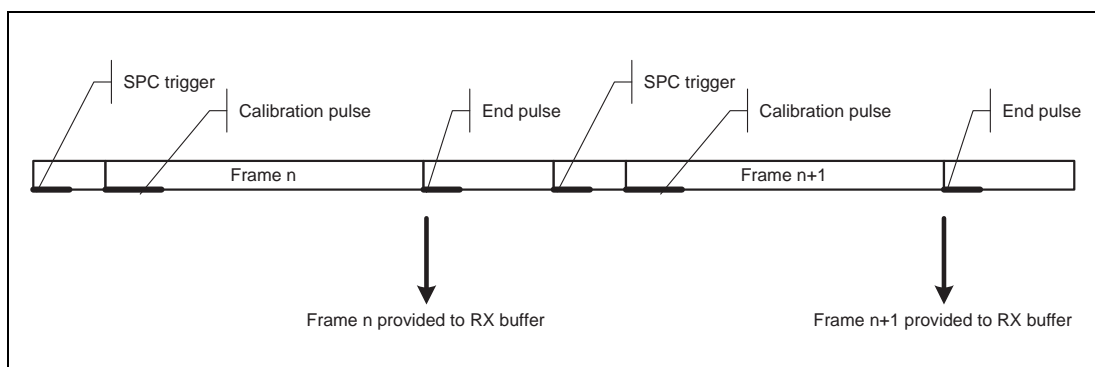


Figure 19.14 Buffer Update in SPC Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

The RSENT module provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples. The CPU needs to calculate the ratio of calibration pulses and/or message length and either accept or discard the message.

In case of variable message length mode, the RSENT module cannot perform this check because the receive timing of the next calibration pulse depends on the next SPC trigger timing.

19.6.2.4 Fast Channel Reception Flow

In **Figure 19.15**, the recommended reception flow for the fast channel message reception buffer is shown.

When using a polling or event driven method, the CPU should only read the setting of the RSENTnCS.FRS bit to check the availability of new fast channel data.

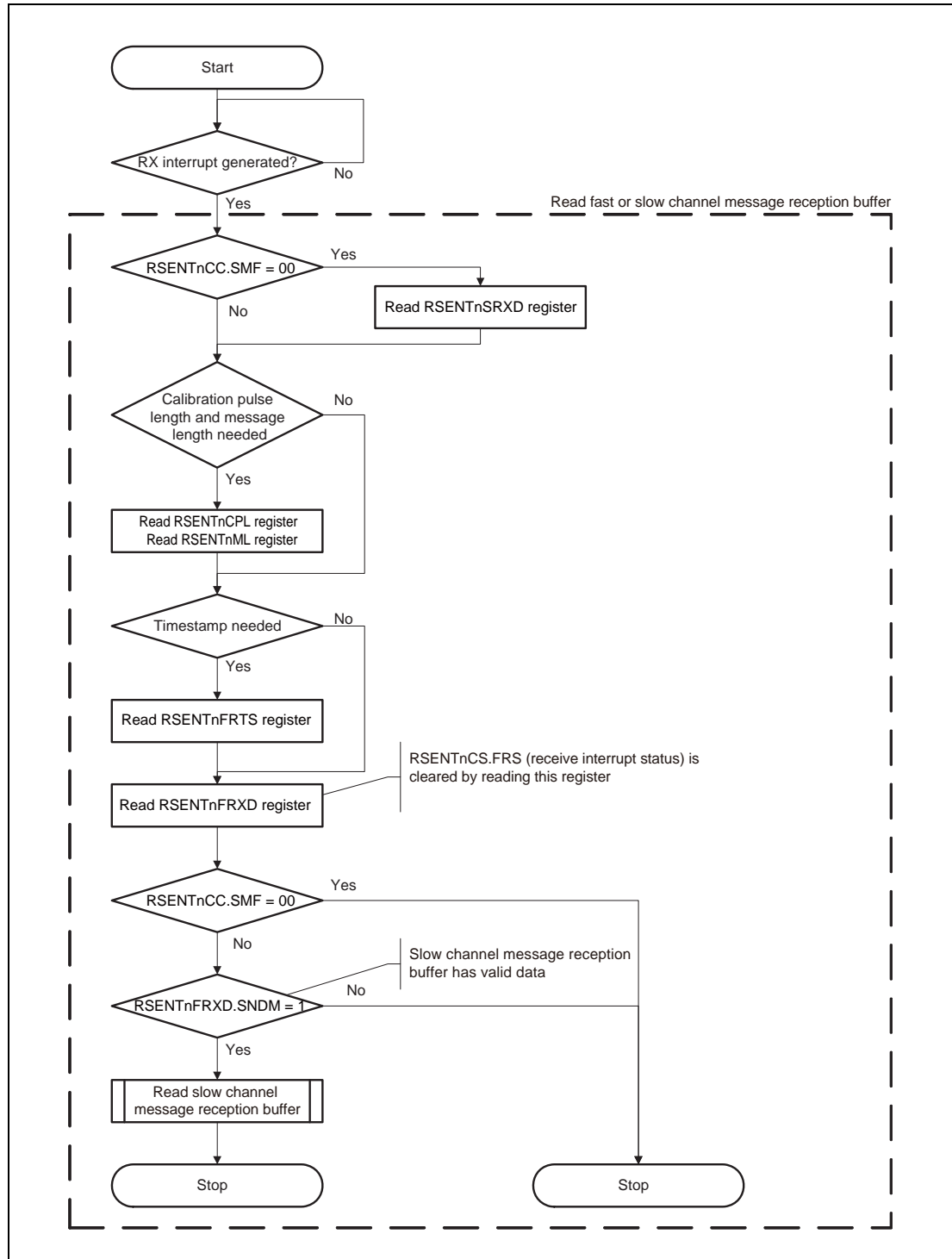


Figure 19.15 Fast Channel Reception Flow

In any case, the CPU should keep the order in reading the receive buffer registers as shown in the flow. The RSENTnFRXD register should be the last register to be accessed.

The handling of the slow channel message reception buffer is described in **Section 19.6.2.6, Slow Channel Reception Flow**.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

19.6.2.5 Slow Channel Message Reception

The RSENT module supports extraction of the slow message out of the fast channel messages by using the status and communications nibble bits 3 and 2 out of the status and communications nibble. In order to enable the slow channel extraction, the CPU should set the RSENTnCC.SMF bits to the expected serial message format.

When no serial message extraction is selected (RSENTnCC.SMF = 00_B), the RSENTnSRXD register becomes part of the fast channel message reception buffer structure (including buffer 2) and RSENTnSRTS register should be ignored. The communications and status nibble is placed in the RSENTnSRXD.IDD bits. Furthermore no slow channel new data and slow channel message lost flags are generated.

In order to receive the slow channel serial message, all fast channel serial messages contributing to a slow channel serial message must be received successfully and the received slow channel serial message must comply with the selected serial message format.

A message lost on the fast channel does not impact the reception on the slow channel.

A slow channel message reception buffer is composed of the slow channel receive timestamp register (RSENTnSRTS) and the slow channel receive data register (RSENTnSRXD).

In opposite to the fast channel message reception buffer, the slow channel message reception buffer does not support a double receive buffer structure; only a single receive buffer structure is available.

Message decoding and assembling is done in a separate register stage.

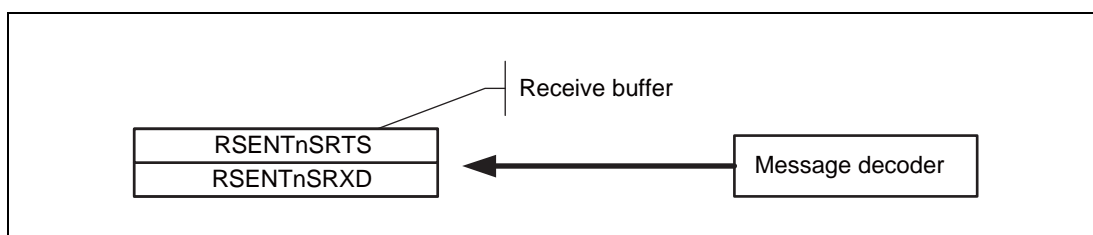


Figure 19.16 Slow Channel Message Reception Buffer

The slow channel message reception buffer is updated at the same time as the fast channel message reception buffer that holds the last status and communications nibble required for the slow channel message. The RSENTnSRXD.SND bit is set to 1 at the same time.

Further updates to the buffer are not carried out until after the RSENTnSRXD.SND bit has been read.

When the receive buffer is holding an unprocessed message (RSENTnSRXD.SND is 1), any further incoming message is lost (the slow channel message reception buffer is not updated) and RSENTnCS.SMS is set to 1.

When the CPU reads the RSENTnSRXD register, RSENTnSRXD.SND is automatically cleared.

The RSENTnSRTS register is updated with the current timestamp counter register value of the last frame contributing to the slow channel message.

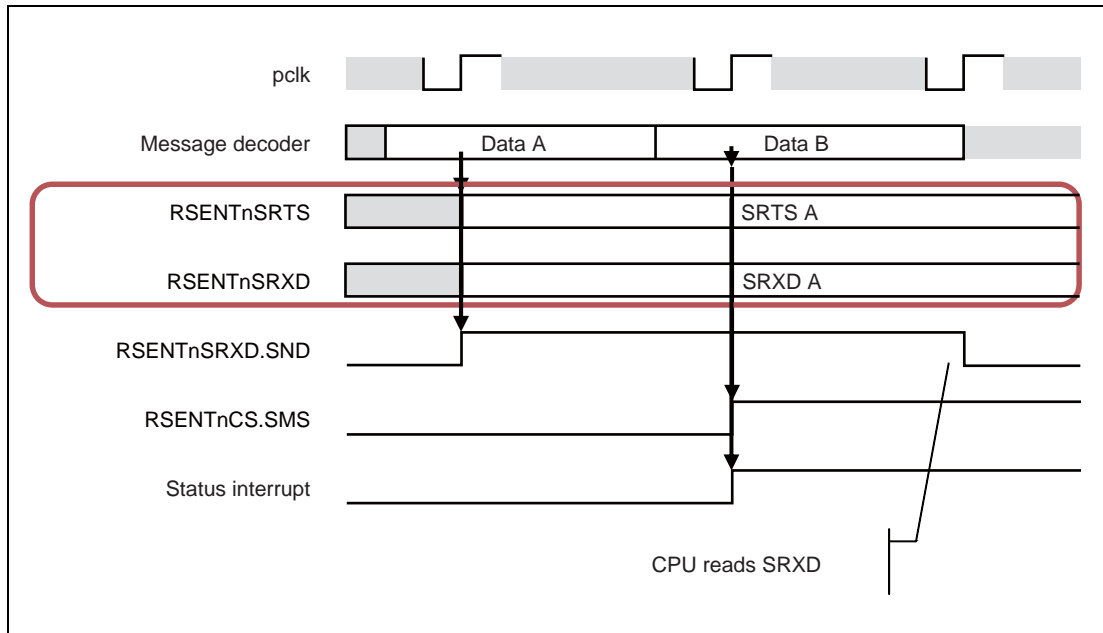


Figure 19.17 Slow Channel Message Reception Buffer Update Timing

The data alignment in the SRDX register depends on the slow channel message format (RSENTnCC.SMF) and the received configuration bit.

Table 19.29 Data Alignment in RSENTnSRXD Register

| RSENTn CC.SMF | RSENTnSRXD.SMGC | RSENTnSRXD.IDD [19:16] | RSENTnSRXD.IDD [15:12] | RSENTnSRXD.IDD [11:8] | RSENTnSRXD.IDD [7:4] | RSENTnSRXD.IDD [3:0] |
|------------------|-----------------|---------------------------|---------------------------|--------------------------|-------------------------|-------------------------|
| 00 _B | Undefined | Undefined | Undefined | Undefined | Undefined | C & S nibble |
| 01 _B | Undefined | Undefined | Undefined | Message ID[3:0] | DATA[7:4] | DATA[3:0] |
| 10 _B | 0 | Message ID[7:4] | Message ID[3:0] | DATA[11:8] | DATA[7:4] | DATA[3:0] |
| 10 _B | 1 | Message ID[3:0] | DATA[15:12] | DATA[11:8] | DATA[7:4] | DATA[3:0] |

19.6.2.6 Slow Channel Reception Flow

In **Figure 19.18**, the recommended reception flow for the slow channel message reception buffer is shown. When the slow channel receive data is required, this process should be executed as part of the fast channel reception flow.

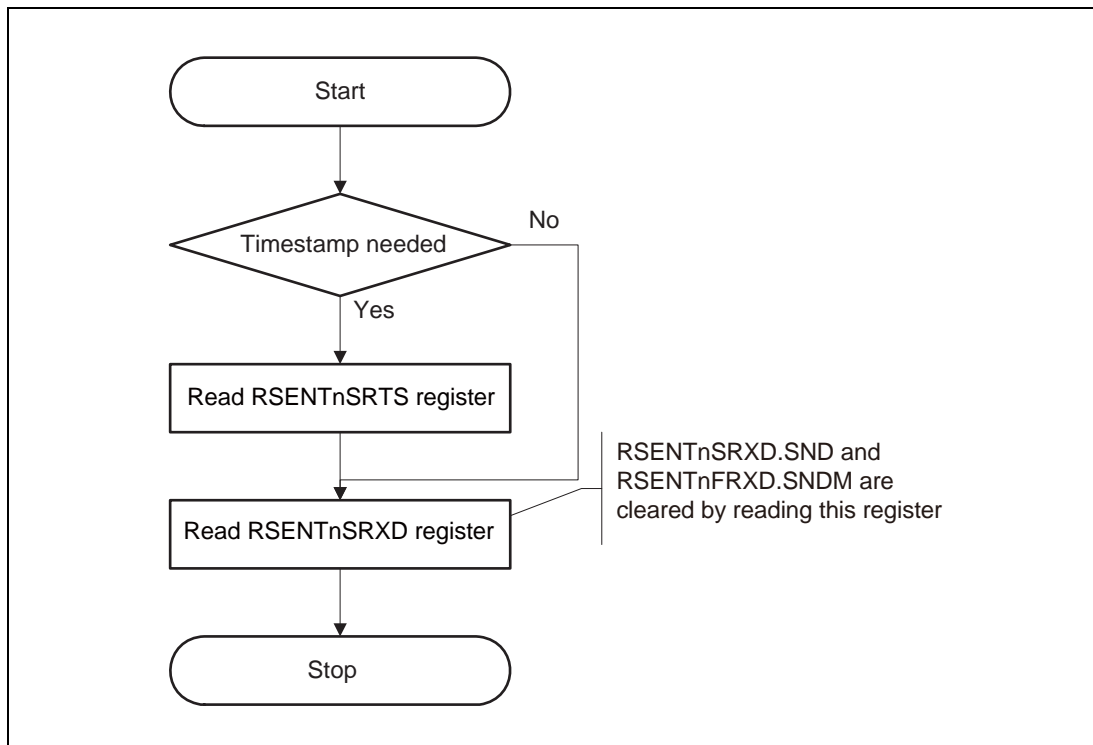


Figure 19.18 Slow Channel Reception Flow

In any case, the CPU should keep the order in reading the slow channel message reception buffer registers as shown in the flow. The RSENTnSRXD.SND bit should be accessed as last.

19.6.2.7 DMA Flow

In case of DMA usage, the start address for the DMA usage and the number of transfers define which part of the receive buffer will be transferred. The RSENTnFRXD register should be the last register to be accessed using a 32 bit access method.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

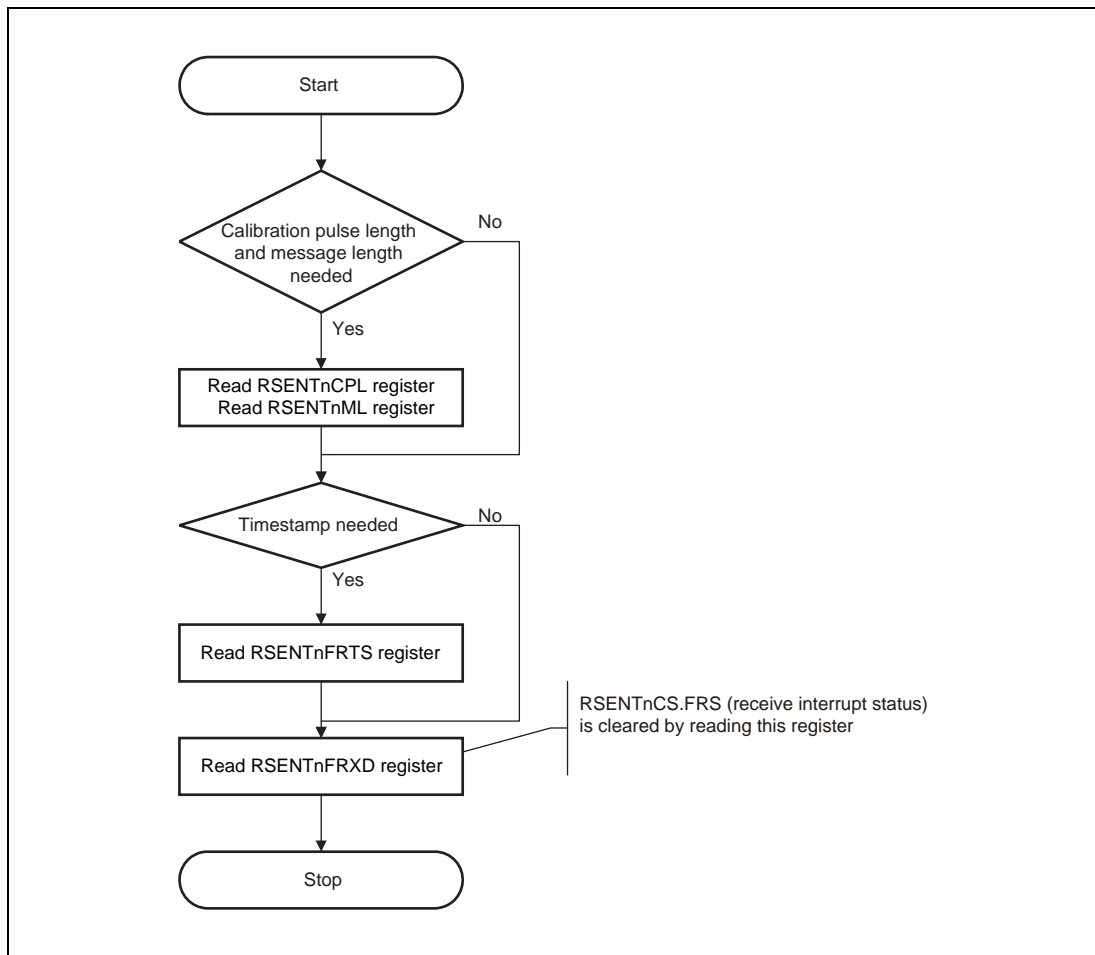


Figure 19.19 DMA Reception Flow

In the software processing, when the transferred data set, the CPU should check the status of the transferred RSENTnFRXD.SNDM bit. If this bit is set to 1, then the user needs to read the slow channel message reception buffer if needed.

19.6.2.8 Error Flagging

The message lost error flag (RSENTnCS.SMS or RSENTnCS.FMS) is set when a new message's diagnostics pass before the previous message is read.

The no response error flag (RSENTnCS.NRS) is set when the CPU has written to RSENTnSPTC.TLL before or during response reception.

The timing at which the fast channel reception error flags (RSENTnCS.CVS, RSENTnCS.CLS, RSENTnCS.FNS, RSENTnCS.FES, and RSENTnCS.FCS) and the slow channel reception error flags (RSENTnCS.SCS, and RSENTnCS.SES) are updated varies with the setting of bits in the communications configuration register (RSENTnCC.SPCE, RSENTnCC.FCM, RSENTnCC.PPC, and RSENTnCC.PPTC).

Table 19.30 and **Table 19.31** list the timings with which the error flags corresponding to each setting are updated.

When a fast channel nibble encoding error or calibration pulse length error is detected, message reception is aborted immediately. No further error flagging for the given message is done. Message decoding starts again after the detection of a valid calibration pulse.

When OPERATION IDLE mode is entered in response to the setting of RSENTnMDC.OMC, error flags of errors for the calibration or the fast channel reception which have been detected in a message being received are not set. A received message is also discarded.

If a fast channel nibble encoding error or calibration pulse length error is detected, entry to OPERATION IDLE mode is immediate.

If a fast channel nibble count error, fast channel CRC error, or calibration pulse length variation error is detected, entry to OPERATION IDLE mode proceeds at the end of the next status and communications nibble.

The fast channel nibble count error flag (RSENTnCS.FNS) is only set when a valid calibration pulse is detected and the subsequent data nibble has a valid length (≥ 12 ticks and ≤ 27 ticks) or no data nibble has been received between two valid calibration pulses.

The fast channel nibble encoding error flag (RSENTnCS.FES) is only set when an encoding error has occurred in a status and communications nibble, CRC nibble, or data nibble.

When the SPC function is enabled (i.e. RSENTnCC.SPCE = 1), the calibration pulse length error flag (RSENTnCS.CLS) is set when the width of a pulse at the expected calibration pulse position is not a valid pulse width for a calibration pulse. When the SPC function is disabled (i.e. RSENTnCC.SPCE = 0), the calibration pulse length error flag (RSENTnCS.CLS) is set when the width of a pulse at a next expected calibration pulse position after the reception of a valid calibration pulse is not a valid pulse width for a calibration pulse.

An additional error flag being set while seeking a valid calibration pulse does not affect the reception of subsequent frames.

Table 19.30 Timing to Set Error Flag (RSENTnCC.SPCE = 0)

| RSENTnCC.SPCE | 0 | | | | | | | |
|---------------|----|---|----|----|----|---|----|----|
| RSENTnCC.FCM | 0 | | | | 1 | | | |
| RSENTnCC.PPC | 0 | | 1 | | 0 | | 1 | |
| RSENTnCC.PPTC | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| RSENTnCS.FCS | EC | x | EC | IM | IM | x | IM | IM |
| RSENTnCS.FES | EC | x | EC | IM | IM | x | IM | IM |
| RSENTnCS.FNS | EC | x | EC | — | — | x | — | — |
| RSENTnCS.SCS | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.SES | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.CLS | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.CVS | EC | x | EC | — | EC | x | EC | — |

Note: EC: End of calibration pulse (on the falling edge of a valid calibration pulse)
IM: Immediately upon detection (immediately when the error is detected)
—: Error flag not detected
x: Setting prohibited

Table 19.31 Timing to Set Error Flag (RSENTnCC.SPCE = 1)

| RSENTnCC.SPCE | 1 | | | | | | | |
|---------------|----|---|----|----|----|---|----|----|
| RSENTnCC.FCM | 0 | | | | 1 | | | |
| RSENTnCC.PPC | 0 | | 1 | | 0 | | 1 | |
| RSENTnCC.PPTC | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| RSENTnCS.FCS | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.FES | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.FNS | — | x | — | — | — | x | — | — |
| RSENTnCS.SCS | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.SES | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.CLS | IM | x | IM | IM | IM | x | IM | IM |
| RSENTnCS.CVS | — | x | — | — | — | x | — | — |

Note: EC: End of calibration pulse (on the falling edge of a valid calibration pulse)
IM: Immediately upon detection (immediately when the error is detected)
—: Error flag not detected
x: Setting prohibited

NOTE

When the sensor stops communications, the reception buffer and the state will not be updated after reception of the last message. Confirm the timeout by software.

19.7 SPC Function

The RSENT module supports an extension of the J2716 specification known as SPC, whereby the RSENT module can pull down the RX line to initiate RSENT message transmission.

The user can configure the polarity of the sent_spc port.

The text below describes the behavior of the sent_spc port with the default settings of RSENTnCC.SOPC. When the value of RSENTnCC.SOPC after a reset is changed, the sent_spc port operates with inverted polarity.

The user can enable or disable the SPC extension by setting the RSENTnCC.SPCE bit. When the RSENTnCC.SPCE bit is set to 0, SPC is disabled. The sent_spc port is driven low by the RSENT module, allowing normal RSENT reception to take place. When the RSENTnCC.SPCE bit is set to 1, SPC is enabled and the sent_spc port can be driven high by the RSENT module to request a frame transmission by the sensor.

The transmission function of the RSENT module is a straightforward PWM function. The purpose of this function is to communicate in direction to the sensor by the output sent_spc. With the sent_spc output, the RSENT module can pull down the signal line by an external transistor. The signal line will be held low for a configured length of tick time specified in the RSENTnSPCT.TLL bits.

The tick time is configured with the RSENTnBRP.TTI bits and the RSENTnBRP.TTF bits which are equal to the transmission tick time. For details, see **Section 19.5.2.2, RX and SPC Tick Settings**.

In a single sensor system, this function can be used to trigger data transmission from the sensor. Further data can be sent to the sensor by varying the trigger pulse length. In a multi sensor system, this function can be used to address a dedicated sensor and request a data transmission.

Once RSENT SPC initialization is complete, transmission can be triggered by writing the trigger pulse width to the RSENTnSPCT.TLL register. When a transmission is triggered, the trigger pulse with the configured length is sent. Then a frame reception is expected. After frame reception was done, a new trigger pulse can be sent.

Writing to RSENTnSPCT.TLL requests a SPC trigger transmission. After writing to RSENTnSPCT.TLL, the CPU should read RSENTnCS.NRS to check whether the previous request was completed or not.

In case RSENTnCS.NRS is set, no SPC trigger is sent and any potentially ongoing reception at this time is aborted. The CPU should clear RSENTnCS.NRS by writing 1'b1 to RSENTnCS.NRC. The CPU can write again to RSENTnSPCT.TLL to request a SPC trigger transmission.

In case RSENTnCS.NRS is not set, the CPU should set a reception timeout counter in software. If a reception occurs before the timeout counter elapses, the user should process the received slow and fast channel data as shown in the fast channel reception flow (**Figure 19.15**) and slow channel reception flow (**Figure 19.18**).

When the timeout counter elapses without any successful reception, the addressed sensor seems not to send any valid response. The CPU should analyze the RSENTnCS register to analyze the reason for no successful reception. A new request can be made considering that when RSENTnCS.NRS gets set no SPC trigger is sent.

Purpose of the timeout function is to define a timeout window for response reception in software.

Figure 19.20 shows a transmission flow with a timeout function implemented in software. The timeout function is optional and can be omitted if not needed.

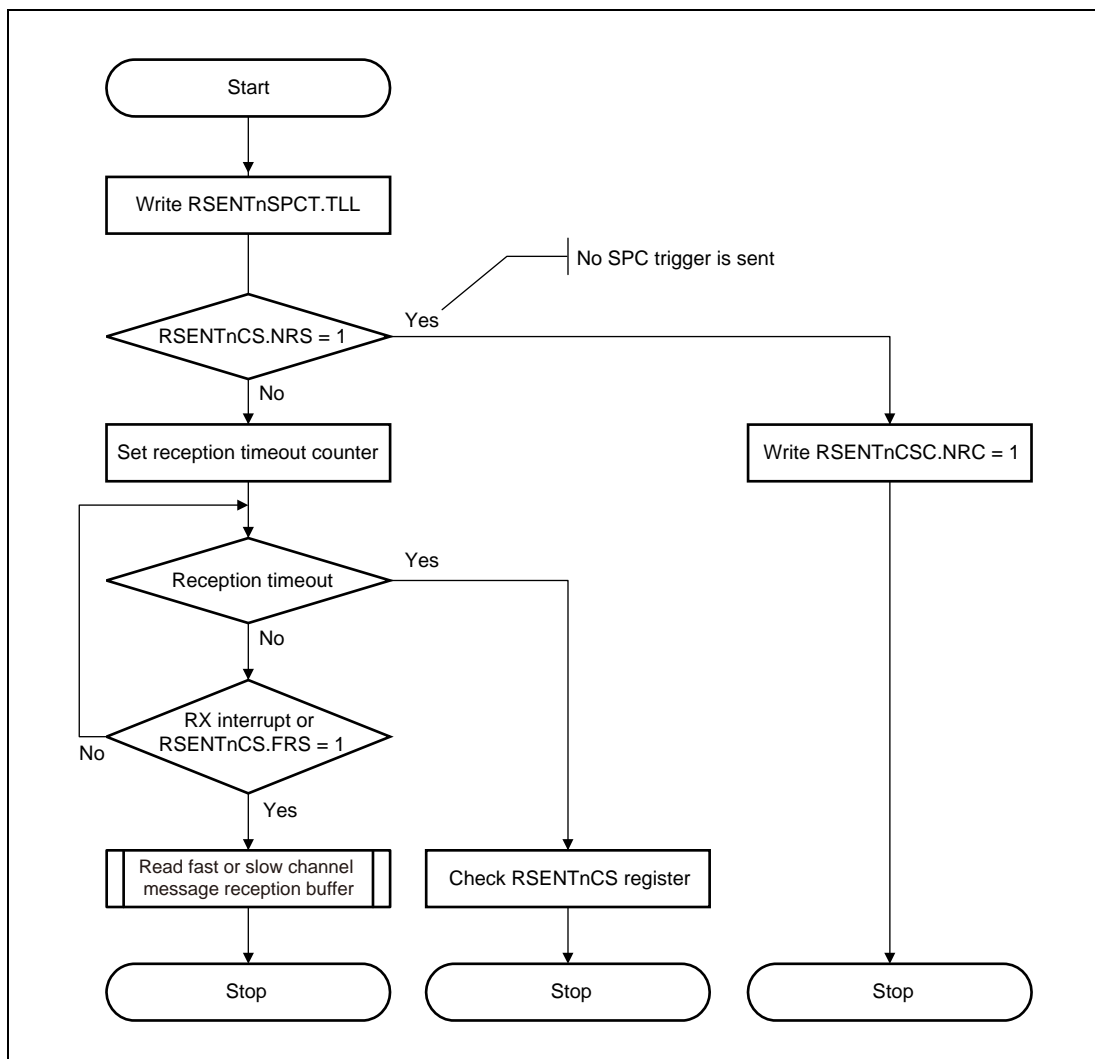


Figure 19.20 Transmission Flow

19.7.1 Multiplexing of the SENTnRX and SENTnSPCO Pin Functions

The SENTnRX input pin and SENTnSPCO output pin functions are assigned to the same multiplexed pin. Two functions can be used on the pin, when it is set for an N-ch open-drain configuration.

Figure 19.21 is a block diagram of multiplexing of the SENTnRX and SENTnSPCO pin functions.

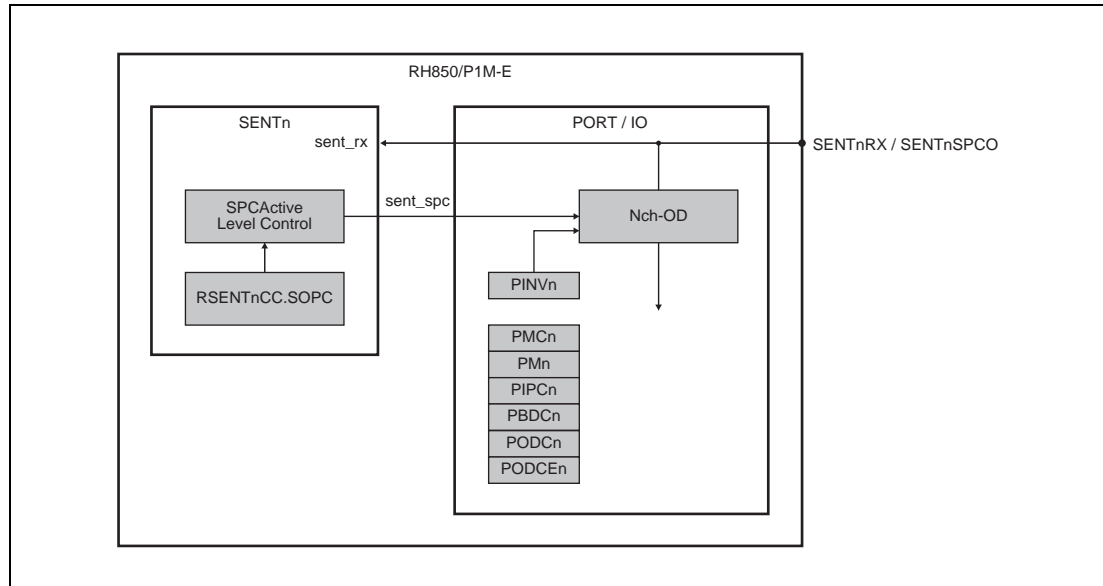


Figure 19.21 Multiplexing of the SENTnRX and SENTnSPCO Pin Functions

Connect the sent_spc signal, which is an RSENT macro output signal, to an N-ch open drain I/O buffer. The RSENT hardware macro is able to control the polarity of sent_spc output and the polarity of the pin output by setting the SOPC bit and the PINVn register of the I/O buffer, respectively. These registers must be set in an appropriate combination.

Table 19.32 lists the register settings when the SENTnRX and SENTnSPCO pin functions share the same pin.

Table 19.32 Setting for Multiplexing of the SENTnRX and SENTnSPCO Pin Functions

| Register Name | Setting | Description |
|---------------|---|---|
| PMcN | Port mode control register | 1 Alternative mode |
| PMn | Port mode register | 0 Output mode (output enabled) |
| PIPcN | Port IP control register | 0 I/O mode is controlled by PMn.PMn_m (software or hardware I/O control) |
| PBDCn | Port bi-direction control register | 1 Bi-direction mode is enabled |
| PODCn | Port open-drain control register | 1 N-ch open drain (PODCn_m = 1 or PODCEn_m = 0) |
| PODCEn | Port open-drain expansion register | 0 N-ch open drain (PODCn_m = 1 or PODCEn_m = 0) |
| PINVn | Port output level inversion register | 0 The pin output level is not inverted (active low when RSENTnCC.SOPC = 1) |
| | | 1 The pin output level is inverted (active high when RSENTnCC.SOPC = 0) |
| PFCn | Port function control register | — Specify an alternative function of the pins. For details, see Table 2.6, Outline of Alternative Mode Selection (PMcN.PMcN_m = 1) . |
| PFCEn | Port function control expansion register | |
| PFCAEn | Port function control addition expansion register | |

19.8 Interrupts and Checks

The RSENT module provides two interrupt lines.

The successful fast channel receive interrupt notifies the CPU that the fast channel message reception buffer was updated and is holding a set of valid received data. Also, the reception status bit is set (RSENTnCS.FRS).

The status interrupt notifies the CPU that at least one of the error flags or message lost flags in the RSENTnCS register is set.

Whether a status flag in the RSENTnCS register is contributing to the generation of an interrupt event or not can be set individually.

The execution of the CRC checks can be disabled for the slow channel and fast channel individually. In case a check is disabled, the CRC of the received message is not checked and the related error flag is never set.

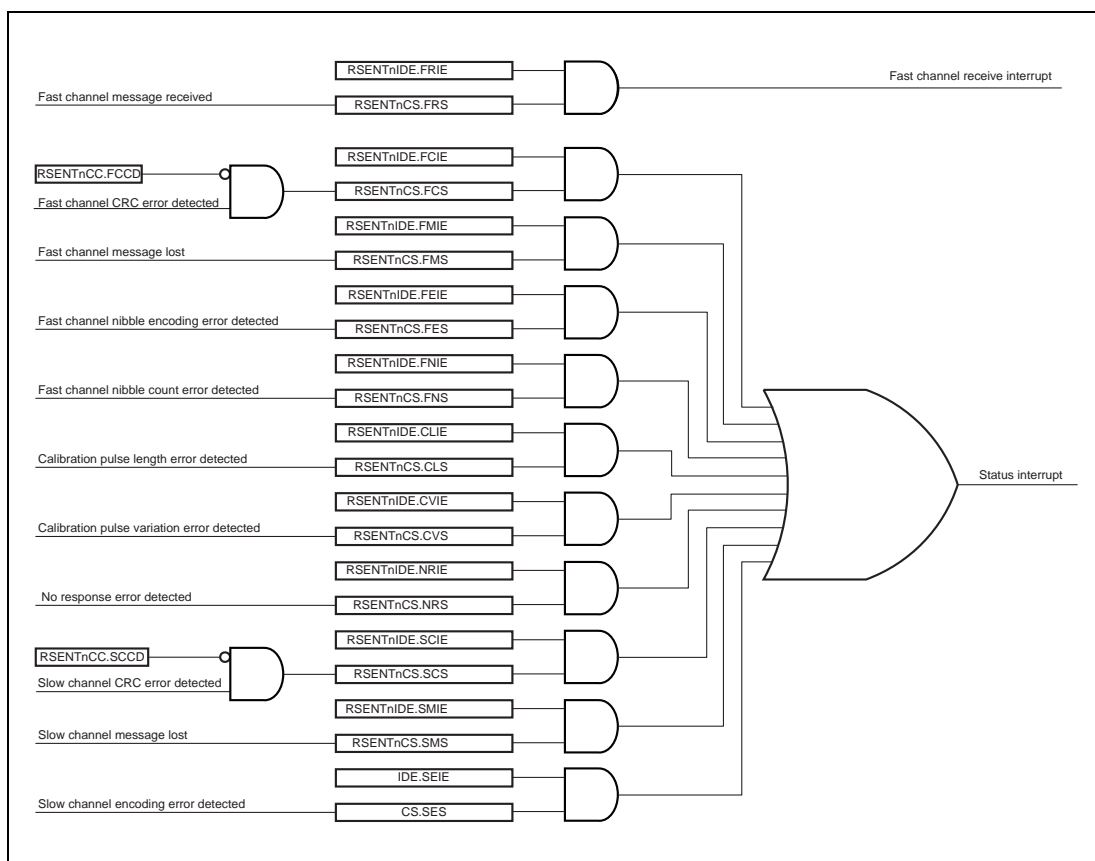


Figure 19.22 Interrupt Structure

Table 19.33 gives an overview about the relationship between set status flags and the buffer update.

Table 19.33 Status Flag Influence to Receive Buffer Behavior

| RSENTnCS | Fast Channel Message Reception Buffer | Slow Channel Message Reception Buffer |
|----------|---------------------------------------|--|
| FRS | Updated | Updated if all status and communications nibbles of slow channel messages are received and RSENTnCS.SES=0 and RSENTnCS.SCS=0 |
| FCS | Not updated | Receive process aborted. Search for new start condition |
| FMS | Message lost | Not impacted |
| FES | Not updated | Receive process aborted. Search for new start condition |
| FNS | Not updated | Receive process aborted. Search for new start condition |
| CLS | Not updated | Receive process aborted. Search for new start condition |
| CVS | Not updated | Receive process aborted. Search for new start condition |
| NRS | Not updated | Receive process aborted. Search for new start condition |
| SCS | Not impacted | Not updated |
| SMS | Not impacted | Message lost |
| SES | Not impacted | Receive process aborted. Search for new start condition |

Section 20 PSI5

The Peripheral Sensor Interface 5 (PSI5) is a function of interface the PSI5 (PSI5 v2.0) standard for sensor. PSI5 is implemented 2 channels.

20.1 Features of RH850/P1M-E PSI5

20.1.1 Number of Channels

This microcontroller has the following number of PSI5 channels.

Table 20.1 Number of Channels

| Product | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|--------------------|-------------------------|-------------------------|
| Number of channels | 2 | 2 |
| Name | PSI5n (n = 0, 1) | PSI5n (n = 0, 1) |

Table 20.2 Index

| Index | Meaning |
|-------|--|
| n | Throughout this section, the individual channels of the PSI5 are identified by the index "n" (n = 0, 1). |

20.1.2 Register Base Address

PSI5n base addresses are listed in the following table. PSI5n register addresses are given as offsets from the base address in general.

Table 20.3 Register Base Address

| Base address Name | Base Address |
|-------------------|------------------------|
| <PSI50_base> | FFE0 0000 _H |
| <PSI51_base> | FFE0 1000 _H |

20.1.3 Clock Supply

Clock supply by and to PSI5n is listed in the following table.

Table 20.4 Clock Supply

| Unit Name | Clock for the Unit | Supply Clock Name |
|-----------|----------------------|-------------------------------------|
| PSI5n | PCLK psi5_com_clk | High-speed peripheral clock CLK_HSB |

20.1.4 Interrupt Requests

The PSI5 can generate the interrupt requests shown in the following table.

Table 20.5 Interrupt Requests

| Unit Interrupt Name | Outline | Interrupt | DMA/DTS Trigger |
|--------------------------|--------------------------|-----------|-----------------|
| PSI50 | | | |
| INTPSI50SI* ¹ | PSI50 status interrupt | 226 | — |
| INTPSI50RI | PSI50 receive interrupt | 227 | 117 |
| INTPSI50TI | PSI50 transfer interrupt | 228 | — |
| PSI51 | | | |
| INTPSI51SI* ¹ | PSI51 status interrupt | 229 | — |
| INTPSI51RI | PSI51 receive interrupt | 230 | 118 |
| INTPSI51TI | PSI51 transfer interrupt | 231 | — |

Note 1. A set of multiple interrupts.

20.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

20.1.6 External Input/Output Signals

The external input/output signals of the PSI5 are listed in the following table.

Table 20.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|------------------|-----|-------------------|----------------------------------|
| PSI50 | | | |
| psi5_rx_data | I | PSI50 data input | Port PSI50DIN |
| psi5_tx_data | O | PSI50 data output | Port PSI50DOUT |
| PSI51 | | | |
| psi5_rx_data | I | PSI51 data input | Port PSI51DIN |
| psi5_tx_data | O | PSI51 data output | Port PSI51DOUT |

20.2 Functions

Features and functions

Among the main functions defined under the PSI5 standard (PSI5 v2.0), this IP supports the following:

- Communication Mode
 - PSI5-A: Asynchronous mode
 - PSI5-P: Synchronous parallel bus mode
 - PSI5-U: Synchronous parallel universal bus mode
 - PSI5-D: Synchronous daisy chain bus mode
 - PSI5-V: Variable time triggered synchronous operation mode
- Sensor to ECU Communication
 - Data can be received up to eight slots.
 - Bit rates: Low speed (125 Kbps); High speed (189 Kbps)
 - Automatic detection of start bits
 - Manchester to binary code conversion
 - Receivable bit length: 10 to 28 bits
 - Serial message frames can be received up to eight slots.
 - Automatic calculation of check bits for both data and serial message frames
 - Stores CRC and parity bits received with the data.
- ECU to Sensor Communication
 - Tooth Gap Mode
 - Pulse Width Mode
 - Automatic detection of start conditions
 - Frame formats 1 to 4 can be used
 - Automatic appending of synchronization bits
 - Automatic appending of CRCs
- PAS compatibility mode
 - 250 Kbps
 - MSB First reception
 - Parity check
- Appending of a timestamp to receive data and serial message frames
 - Master and slave function for synchronization of timestamp

Table 20.7 List of Operation Modes

| Operation Mode | Normal Mode | PAS Compatibility Mode |
|-----------------------------|--|------------------------|
| Bit rate | 125 Kbps/189 Kbps | 250 Kbps |
| Data word length | 10 to 28 bits | 8 to 24 bits |
| Data direction | LSB first | MSB first |
| Data transmission parameter | | |
| Value of bit time (TYP.) | 8 μ s/5.3 μ s | 4 μ s |
| Value of gap time (MIN.) | 8.4 μ s/5.6 μ s | 2 μ s |
| Communication Mode | PSI5-A, PSI5-S, PSI5-U, PSI5-D, PSI5-V | PSI5-A only |

20.2.1 Block Configuration

Figure 20.1 is a block diagram of PSI5n.

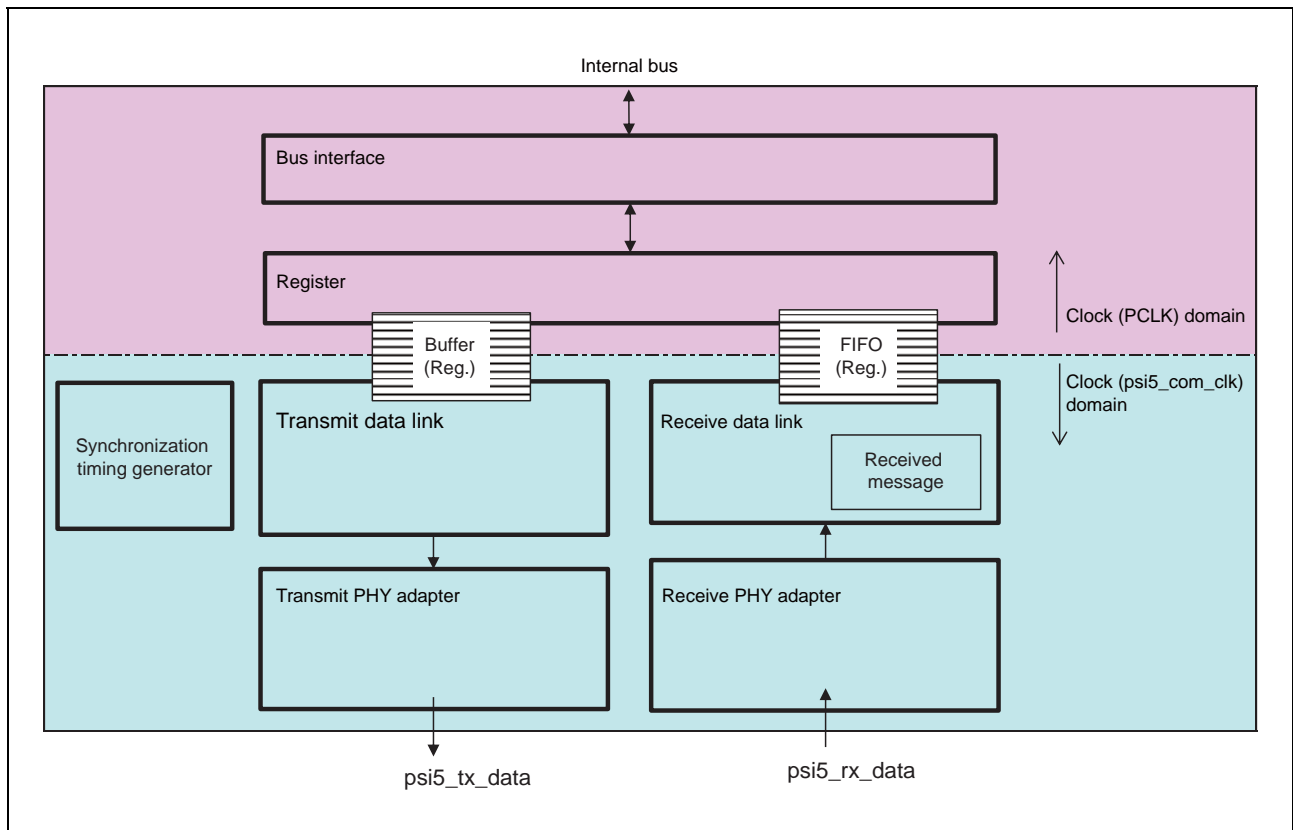


Figure 20.1 Block Diagram of PSI5

20.3 Registers

The PSI5n is controlled and operated by the registers listed in the table below.

For PSI5n base address <PSI5n_base>, see **Section 20.1.2, Register Base Address**.

Table 20.8 PSI5n Register Overview (1/2)

| Register Name | Symbol | Address |
|---|-------------------|----------------------------------|
| PSI5 channel control register | PSI5nCHCTRL | <PSI5n_base> + 0000 _H |
| PSI5 IP timer control register | PSI5nIPTIMERCTRL | <PSI5n_base> + 0010 _H |
| PSI5 IP timer counter | PSI5nIPTIMER | <PSI5n_base> + 0014 _H |
| PSI5 operating mode/communication mode register | PSI5nOPMCOMM | <PSI5n_base> + 0020 _H |
| PSI5 operating-mode bit rate register | PSI5nOPMBITRATE | <PSI5n_base> + 0024 _H |
| PSI5 operating-mode cycle time register | PSI5nOPMCYCT | <PSI5n_base> + 0028 _H |
| PSI5 interrupt status register | PSI5nPSI5INT | <PSI5n_base> + 0030 _H |
| PSI5 receive data emulation register | PSI5nEMRXDATA | <PSI5n_base> + 0040 _H |
| PSI5 receive data status emulation register | PSI5nEMRXDST | <PSI5n_base> + 0044 _H |
| PSI5 receive data IP timer emulation register | PSI5nEMRXDTIM | <PSI5n_base> + 0048 _H |
| PSI5 receive data FIFO emulation register | PSI5nEMRXDFIFO | <PSI5n_base> + 004C _H |
| PSI5 receive-message receive message emulation register | PSI5nEMRXMRXMSG | <PSI5n_base> + 0050 _H |
| PSI5 receive-message channel receive status emulation register | PSI5nEMRXMRXST | <PSI5n_base> + 0054 _H |
| PSI5 receive-message channel receive timestamp emulation register | PSI5nEMRXMRXTIM | <PSI5n_base> + 0058 _H |
| PSI5 receive-message channel FIFO emulation register | PSI5nEMRXMFIFO | <PSI5n_base> + 005C _H |
| PSI5 transmission setting register | PSI5nTXSETTING | <PSI5n_base> + 0080 _H |
| PSI5 synchronization control register | PSI5nSYNCCTRL | <PSI5n_base> + 0084 _H |
| PSI5 transmission status register | PSI5nTXST | <PSI5n_base> + 0088 _H |
| PSI5 transmission status clear register | PSI5nTXSTCLR | <PSI5n_base> + 008C _H |
| PSI5 transmission status interrupt enable register | PSI5nTXSTINTEN | <PSI5n_base> + 0090 _H |
| PSI5 transmit data control register | PSI5nTXDCTRL | <PSI5n_base> + 0094 _H |
| PSI5 transmit data register | PSI5nTXDATA | <PSI5n_base> + 0098 _H |
| PSI5 receive sampling setting register | PSI5nRXSPLSET | <PSI5n_base> + 0100 _H |
| PSI5 receive slot 1 setting register | PSI5nRXS1SET | <PSI5n_base> + 0108 _H |
| PSI5 receive slot 2 setting register | PSI5nRXS2SET | <PSI5n_base> + 010C _H |
| PSI5 receive slot 3 setting register | PSI5nRXS3SET | <PSI5n_base> + 0110 _H |
| PSI5 receive slot 4 setting register | PSI5nRXS4SET | <PSI5n_base> + 0114 _H |
| PSI5 receive slot 5 setting register | PSI5nRXS5SET | <PSI5n_base> + 0118 _H |
| PSI5 receive slot 6 setting register | PSI5nRXS6SET | <PSI5n_base> + 011C _H |
| PSI5 receive slot 7 setting register | PSI5nRXS7SET | <PSI5n_base> + 0120 _H |
| PSI5 receive slot 8 setting register | PSI5nRXS8SET | <PSI5n_base> + 0124 _H |
| PSI5 receive data register | PSI5nRXDATA | <PSI5n_base> + 0128 _H |
| PSI5 receive data status register | PSI5nRXDST | <PSI5n_base> + 012C _H |
| PSI5 receive data IP timer register | PSI5nRXDTIM | <PSI5n_base> + 0130 _H |
| PSI5 receive data FIFO register | PSI5nRXDFIFO | <PSI5n_base> + 0134 _H |
| PSI5 receive module status register | PSI5nRXMODST | <PSI5n_base> + 0138 _H |
| PSI5 receive module status clear register | PSI5nRXMODSTCLR | <PSI5n_base> + 013C _H |
| PSI5 receive module status interrupt enable register | PSI5nRXMODSTINTEN | <PSI5n_base> + 0140 _H |
| PSI5 receive message channel setting register | PSI5nRXMSET | <PSI5n_base> + 0180 _H |
| PSI5 receive-message receive message register | PSI5nRXMRXMSG | <PSI5n_base> + 0184 _H |

Table 20.8 PSI5n Register Overview (2/2)

| Register Name | Symbol | Address |
|--|------------------|----------------------------------|
| PSI5 receive-message channel receive status register | PSI5nRXMRXST | <PSI5n_base> + 0188 _H |
| PSI5 receive-message channel receive timestamp register | PSI5nRXMRXTIM | <PSI5n_base> + 018C _H |
| PSI5 receive-message channel FIFO register | PSI5nRXMFIFO | <PSI5n_base> + 0190 _H |
| PSI5 receive-message channel module status register | PSI5nRXMMST | <PSI5n_base> + 0194 _H |
| PSI5 receive-message channel module status clear register | PSI5nRXMMSTCLR | <PSI5n_base> + 0198 _H |
| PSI5 receive-message channel module status interrupt enable register | PSI5nRXMMSTINTEN | <PSI5n_base> + 019C _H |
| PSI5 timestamp function mode selection register | PSI5TSSEL | <PSI50_base> + 3000 _H |

20.3.1 PSI5nCHCTRL — PSI5 Channel Control Register

This register controls the channel operation.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0000_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CHEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 20.9 PSI5nCHCTRL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | CHEN | <p>Channel Enable</p> <p>0: Channel disabled</p> <p>1: Channel enabled</p> <p>In synchronous mode, the PSI5 starts to send a synchronization pulse after enabling the channel. In asynchronous mode, the PSI5 sets psi5_tx_data according to PSI5nTXSETTING.DEFTXVAL.</p> <p>When the channel is enabled, writing to the following registers is disabled: PSI5nOPMCOMM, PSI5nOPMBITRATE, PSI5nOPMCYCT, PSI5nTXSETTING, PSI5nRXSPLSET, PSI5nRXSmSET, and PSI5nRXMSET</p> <p>After writing to this bit, check that the setting is reflected by reading the bit by software.</p> |

CAUTION

The following registers must be set only when channel operation is enabled (PSI5nCHCTRL.CHEN = 1).

- PSI5nRXMODSTINTEN.RXDEXISTINTEN = 1
- PSI5nTXSTINTEN.TXEMPTYINTEN = 1
- PSI5nTXDATA.TXDATA

20.3.2 PSI5nIPTIMERCTRL — PSI5 IP Timer Control Register

This register controls the IP timer.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0010_H

Value after reset: 0001 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|---------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSTSLV |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TSCLR | — | — | — | — | — | — | — | IPTIMEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R | R/W |

Table 20.10 PSI5nIPTIMERCTRL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 17 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 16 | MSTSLV | Specifies timer master/slave mode. 0: Slave mode is selected 1: Master mode is selected <ul style="list-style-type: none"> In slave mode: <p>The IP timer counts up according to the count-up timing signal (psi5_ts_tick_in) input through the master channel. The value of the PSI5nIPTIME register is cleared when the TSCLR bit is set or in response to the input of the clearing timing signal (psi5_ts_clr_in) through the master channel.</p> In master mode: <p>IPTIMER counts up in response to the clock generated by the internal baud rate generator. PSI5 outputs the count-up timing signal (psi5_ts_tick_out). Setting the TSCLR bit to 1 clears the values in PSI5nIPTIMER (the clearing timing signal (psi5_ts_clr_out) is output).</p> |
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | TSCLR | IP Timer Clear Trigger 0: The IP timer (PSI5nIPTIME register) is not cleared. 1: The IP timer (PSI5nIPTIME register) is cleared. The TSCLR bit setting is valid when the MSTSLV bit is set to 1. Read value is always 0. |
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | IPTIMEN | IP Timer Enable 0: IP timer is disabled. 1: IP timer is enabled. In slave mode, set the IPTIMEN bit after enabling the IP timer of the master channel. |

20.3.3 PSI5nIPTIMER — PSI5 IP Timer Counter

This is a counter for the IP timer.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0014_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| COUNTVAL[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNTVAL[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 20.11 PSI5nIPTIMER Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | COUNTVAL | In master mode, this timer counts up in response to the internal baud rate clock. In slave mode, this timer counts up in response to the input of the count-up timing signal (psi5_ts_tick_in) through the master channel. When writing to this register, the value is reflected after several cycles. |

20.3.4 PSI5nOPMCOMM — PSI5 Operating Mode/Communication Mode Register

This register sets the communication mode.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0020_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | COMMODE | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 20.12 PSI5nOPMCOMM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | COMMODE | Specify communication mode. 000 _B : Asynchronous mode 001 _B : Synchronous Parallel Bus Mode 010 _B : Synchronous Universal Bus Mode 011 _B : Synchronous Daisy Chain Bus Mode 100 _B : Variable Time Triggered Synchronous Operation Mode Other than above: Setting prohibited |

20.3.5 PSI5nOPMBITRATE — PSI5 Operating-Mode Bit Rate Register

This register sets a bit rate.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0024_H

Value after reset: 0000 029F_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | BITRATECNT | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 20.13 PSI5nOPMBITRATE Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 to 0 | BITRATECNT | <p>Specify the 1-bit clock count considering communication clock tolerance.</p> <ul style="list-style-type: none"> For transfer at 80 MHz (12.5 ns) and 125 Kbps (clock tolerance: 5%: 8.4 μs) psi5_com_clk is used as the counter clock. Counting cycle = 8.4 μs/12.5 ns = 672 = 2A0_H Setting value: 2A0-001_H = 29F_H <p>This value is used to secure the bit length while no edge is detected in bit judgment.</p> |

20.3.6 PSI5nOPMCYCT — PSI5 Operating-Mode Cycle Time Register

This register sets a count value.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0028_H

Value after reset: 0000 9C3F_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | TTTTCNT | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TTTTCNT | | | | | | | | | | | | | | | |
| Value after reset | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 20.14 PSI5nOPMCYCT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 20 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 19 to 0 | TTTTCNT | Specify the count value. Specify the count value for the SYNC signal's period. Use psi5_com_clk as the count clock. The default value is 500 μs when counting at 80 MHz (T = 12.5 ns). Count cycle: 500 μs / 12.5 ns = 40000 = 9C40 _H Setting value: 9C40 _H -001 _H = 9C3F _H |

20.3.7 PSI5nPSI5INT — PSI5 Interrupt Status Register

This register indicates the status of an interrupt that occurred in the PSI5.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0030_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|------------------------|----------------|-----------------|------------------|----------------|-----------------|--------------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | INT_ SYNCED | INT_ SYNCST | INT_ TXDEMP PTY |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | INT_RX DSCNF ERR | INT_RX DERR | INT_RX DFOVF | INT_RX DEXIST | INT_RX MERR | INT_RX MFOVF | INT_RX MEXIST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.15 PSI5nPSI5INT Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 31 to 19 | Reserved | When read, the value after reset is read. |
| 18 | INT_SYNCED | INT_SYNCED interrupt generation 0: Was not generated 1: Generated |
| 17 | INT_SYNCST | INT_SYNCST interrupt generation 0: Was not generated 1: Generated |
| 16 | INT_TXDEMP TY | INT_TXDEMP interrupt generation 0: Was not generated 1: Generated |
| 15 to 7 | Reserved | When read, the value after reset is read. |
| 6 | INT_RXDSCNF ERR | INT_RXDSCNFERR interrupt generation 0: Was not generated 1: Generated |
| 5 | INT_RXDERR | INT_RXDERR interrupt generation 0: Was not generated 1: Generated |
| 4 | INT_RXDFOVF | INT_RXDFOVF interrupt generation 0: Was not generated 1: Generated |
| 3 | INT_RXDEXIST | INT_RXDEXIST interrupt generation 0: Was not generated 1: Generated |
| 2 | INT_RXMERR | INT_RXMERR interrupt generation 0: Was not generated 1: Generated |
| 1 | INT_RXMFOVF | INT_RXMFOVF interrupt generation 0: Was not generated 1: Generated |
| 0 | INT_RXMEXIST | INT_RXMEXIST interrupt generation 0: Was not generated 1: Generated |

NOTE

For details about interrupts in the PSI5, see **Section 20.4, Interrupt**.

20.3.8 PSI5nEMRXDATA — PSI5 Receive Data Emulation Register

This is a mirror register of the receive data register (PSI5nRXDATA).

For details on PSI5nRXDATA, see **Section 20.3.25, PSI5nRXDATA — PSI5 Receive Data Register**.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0040_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RXDATA | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXDATA | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.16 PSI5nEMRXDATA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | RXDATA | Mirror data of PSI5nRXDATA. Data is not updated even if it is read. |

20.3.9 PSI5nEMRXDST — PSI5 Receive Data Status Emulation Register

This is a mirror register of the receive data status register (PSI5nRXDST).

For details on PSI5nRXDST, see **Section 20.3.26, PSI5nRXDST — PSI5 Receive Data Status Register**.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0044_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|--------|----|-----------|----|----|----|----|----|----|--------------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | RXCHKD | | RXSLOTNUM | | | | — | — | — | RXSTAT US | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.17 PSI5nEMRXDST Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 30 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 8 | RXCHKD | These bits act as mirrors of the PSI5nRXDST.RXCHKD bits. Data is not updated even if it is read. |
| 7 to 4 | RXSLOTNUM | These bits act as mirrors of the PSI5nRXDST.RXSLOTNUM bits. Data is not updated even if it is read. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | RXSTATUS | This bit acts as a mirror of the PSI5nRXDST.RXSTATUS bit. Data is not updated even if it is read. |

20.3.10 PSI5nEMRXDTIM — PSI5 Receive Data IP Timer Emulation Register

This is a mirror register of the receive data IP timer register (PSI5nRXDTIM).

For details on PSI5nRXDTIM, see **Section 20.3.27, PSI5nRXDTIM — PSI5 Receive Data IP Timer Register**.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0048_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | RXDTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXDTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.18 PSI5nEMRXDTIM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 0 | RXDTIM | Mirror data of the PSI5nRXDTIM register. Data is not updated even if it is read. |

20.3.11 PSI5nEMRXDFIFO — PSI5 Receive Data FIFO Emulation Register

This is a mirror register of the receive data FIFO register (PSI5nRXDFIFO).

For details on PSI5nRXDFIFO, see **Section 20.3.28, PSI5nRXDFIFO — PSI5 Receive Data FIFO Register**.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 004C_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | RXDST | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXDST | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.19 PSI5nEMRXDFIFO Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | RXDST | Mirror data of the PSI5nRXDFIFO register. Data is not updated even if it is read. |

20.3.12 PSI5nEMRXMRXMSG — PSI5 Receive-Message Receive Message Emulation Register

This is a mirror register of the receive-message receive message register (PSI5nRXMRXMSG).

For details on PSI5nRXMRXMSG, see **Section 20.3.33, PSI5nRXMRXMSG — PSI5 Receive-Message Receive Message Register.**

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0050_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|
| | CONFI GBIT | — | — | — | — | — | — | — | SERIALID | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DATAFIELD | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.20 PSI5nEMRXMRXMSG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 | CONFIGBIT | This bit acts as a mirror of the PSI5nRXMRXMSG.CONFIGBIT bit. Data is not updated even if it is read. |
| 30 to 24 | Reserved | When read, the value after reset is read. |
| 23 to 16 | SERIALID | These bits act as mirrors of the PSI5nRXMRXMSG.SERIALID bits. Data is not updated even if it is read. |
| 15 to 0 | DATAFIELD | These bits act as mirrors of the PSI5nRXMRXMSG.DATAFIELD bits. Data is not updated even if it is read. |

20.3.13 PSI5nEMRXMRXST — PSI5 Receive-Message Channel Receive Status Emulation Register

This is a mirror register of the receive-message channel receive status register (PSI5nRXMRXST).

For details on PSI5nRXMRXST, see **Section 20.3.34, PSI5nRXMRXST — PSI5 Receive-message Channel Receive Status Register**.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0054_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|-------|----|----|----|----|---------|----|----|----|----|----|--------|----|----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RXSYNC | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | RXCRC | | | | | SLOTNUM | | | | | — | — | — | RXSTATUS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.21 PSI5nEMRXMRXST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 30 to 19 | Reserved | When read, the value after reset is read. |
| 18 to 16 | RXSYNC | These bits act as mirrors of the PSI5nRXMRXST.RXSYNC bits. Data is not updated even if it is read. |
| 15, 14 | Reserved | When read, the value after reset is read. |
| 13 to 8 | RXCRC | These bits act as mirrors of the PSI5nRXMRXST.RXCRC bits. Data is not updated even if it is read. |
| 7 to 4 | SLOTNUM | These bits act as mirrors of the PSI5nRXMRXST.SLOTNUM bits. Data is not updated even if it is read. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | RXSTATUS | This bit acts as a mirror of the PSI5nRXMRXST.RXSTATUS bit. Data is not updated even if it is read. |

20.3.14 PSI5nEMRXMRXTIM — PSI5 Receive-Message Channel Receive Timestamp Emulation Register

This is a mirror register of the receive-message channel receive timestamp register (PSI5nRXMRXTIM).

For details on PSI5nRXMRXTIM, see **Section 20.3.35, PSI5nRXMRXTIM — PSI5 Receive-message Channel Receive Timestamp Register.**

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0058_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RXMTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXMTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.22 PSI5nEMRXMRXTIM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 0 | RXMTIM | Mirror data of the PSI5nRXMRXTIM register. Data is not updated even if it is read. |

20.3.15 PSI5nEMRXMFIFO — PSI5 Receive-Message Channel FIFO Emulation Register

This is a mirror register of the receive-message channel FIFO register (PSI5nRXMFIFO).

For details on PSI5nRXMFIFO, see **Section 20.3.36, PSI5nRXMFIFO — PSI5 Receive-message Channel FIFO Register**.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 005C_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | RXMFIFO | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXMFIFO | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.23 PSI5nEMRXMFIFO Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | RXMFIFO | Mirror data of the PSI5nRXMFIFO register. Data is not updated even if it is read. |

20.3.16 PSI5nTXSETTING — PSI5 Transmission Setting Register

This register specifies transmission settings.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0080_H

Value after reset: 1067 47F7_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|--------------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | LONGCNT | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PHY MODE | DEF TXVAL | SHORTCNT | | | | | | | | | | | | | |
| Value after reset | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 20.24 PSI5nTXSETTING Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | LONGCNT | Set the allowable upper limit of “long” synchronization pulse width when synchronous signal is output in the Pulse Width method. The upper limit when psi5_com_clk is at 80 MHz is 52.5 μs. Counting cycle: 52.5 μs / 12.5 ns = 4200 = 1068 _H Setting value: 1068 _H - 0001 _H = 1067 _H |
| 15 | PHYMODE | Specifies synchronous pulse PHY layer mode. 0: Tooth Gap method is selected. 1: Pulse Width method is selected. When the Pulse Width method is selected, the pulse width should comply with the value within the pulse width specified by the SHORTCNT bits in the Tooth Gap method. |
| 14 | DEFTXVAL | Specifies the initial value of synchronization pulse when data is not transmitted. 0: (0) is transmitted. 1: (1) is transmitted. |
| 13 to 0 | SHORTCNT | Set the allowable upper limit of “short” synchronization pulse width when synchronous signal is output in the Pulse Width or Tooth Gap method. The upper limit when psi5_com_clk is at 80 MHz is 25.5 μs. Counting cycle: 25.5 μs / 12.5 ns = 2040 = 07F8 _H Setting value: 07F8 _H - 0001 _H = 07F7 _H |

CAUTION

- The PSI5nTXSETTING.SHORTCNT bit must be set to 1 or higher.
- The setting value below should be used.
PSI5nOPMCYCT.TTTTCNT > PSI5nTXSETTING.LONGCNT > PSI5nTXSETTING.SHORTCNT

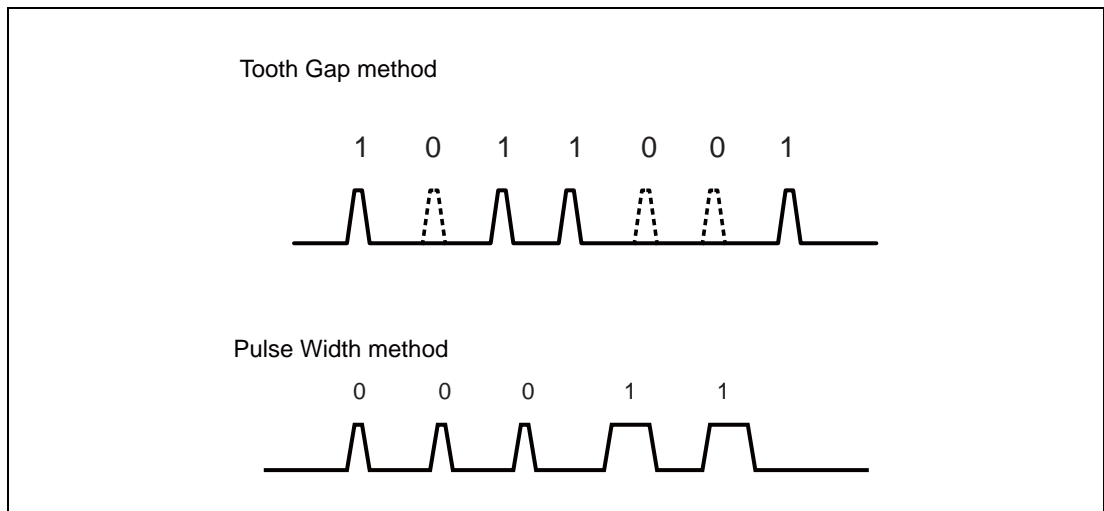


Figure 20.2 Pulse Width Method and Tooth Gap Method

20.3.17 PSI5nSYNCCTRL — PSI5 Synchronization Control Register

This register controls the start trigger in variable time triggered synchronous operation mode.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0084_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VALTIM SYNC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 20.25 PSI5nSYNCCTRL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | VALTIMSYNC | Synchronization signal send trigger in variable time triggered synchronous operation mode 0: Not effective. 1: The synchronization pulse is issued. This bit is always 0 when it is read. Do not write 1 to the VALTIMSYNC bit in any modes other than variable time triggered synchronous operation mode. |

20.3.18 PSI5nTXST — PSI5 Transmission Status Register

This register indicates the transmission status.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0088_H

Value after reset: 0000 0001_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|------------|------------|----|----|----|----|----|----|----|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | SYN CED | SYN CST | — | — | — | — | — | — | — | TXD EMPTY |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.26 PSI5nTXST Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 10 | Reserved | When read, the value after reset is read. |
| 9 | SYNCED | Indicates synchronization signal transmit complete. 0: Synchronization signal transmit is not started or is incomplete. 1: Synchronization signal transmit is complete. When the SYNCED bit is set, the INT_SYNCED interrupt is asserted. The SYNCED bit is set even when 0 is sent in the Tooth Gap method. |
| 8 | SYNCST | Indicates start of synchronization signal transmit 0: Synchronization signal is not sent. 1: Synchronization signal is started to send. When the SYNCST bit is set, the INT_SYNCST interrupt is asserted. The SYNCST bit is set even when 0 is sent in the Tooth Gap method. |
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | TXDEEMPTY | Indicates Tx data buffer empty status. 0: Transmit data buffer is not empty 1: Transmit data buffer is empty When the TXDEEMPTY bit is set, the INT_TXDEEMPTY interrupt is asserted. |

20.3.19 PSI5nTXSTCLR — PSI5 Transmission Status Clear Register

This register clears the setting in the PSI5nTXST register.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 008C_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---------------|---------------|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | SYNC EDCLR | SYNC STCLR | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R | R |

Table 20.27 PSI5nTXSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | SYNCEDCLR | Clears the value of the PSI5nTXST.SYNCED bit. 0: No operation is done. The read value is always 0. 1: Clears the value of the PSI5nTXST.SYNCED bit |
| 8 | SYNCSTCLR | Clears the value of the PSI5nTXST.SYNCST bit. 0: No operation is done. The read value is always 0. 1: Clears the value of the PSI5nTXST.SYNCST bit. |
| 7 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

20.3.20 PSI5nTXSTINTEN — PSI5 Transmission Status Interrupt Enable Register

This register controls transmission status interrupts.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0090_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|---------------------|---------------------|----|----|----|----|----|----|----|----------------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | SYNC EDINT EN | SYNC STINT EN | — | — | — | — | — | — | — | TXD EMPTY NTEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R | R | R | R | R | R | R | R/W |

Table 20.28 TXSINTEN Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 31 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | SYNCED INTEN | Controls the INT_SYNCED interrupt. 0: Interrupt disabled (Masked) 1: Interrupt enabled |
| 8 | SYNCSTINTEN | Controls the INT_SYNCST interrupt. 0: Interrupt disabled (Masked) 1: Interrupt enabled |
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TXDEEMPTYINT EN | Controls the INT_TXDEEMPTY interrupt. 0: Interrupt disabled (Masked) 1: Interrupt enabled |

20.3.21 PSI5nTXDCTRL — PSI5 Transmit Data Control Register

This register controls transmission status interrupts.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0094_H

Value after reset: 0000 0001_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | FRMFORMAT | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 20.29 PSI5nTXDCTRL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | FRMFORMAT | Specify the Tx frame format. 001 _B : Frame 1 (Short) 010 _B : Frame 2 (Long: 4-bit data or 8-bit data) 011 _B : Frame3 (XLong) 100 _B : Frame4 (XXLong) Other than above: Setting prohibited |

20.3.22 PSI5nTXDATA — PSI5 Transmit Data Register

This register stores transmit data.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0098_H

Value after reset: Undefined

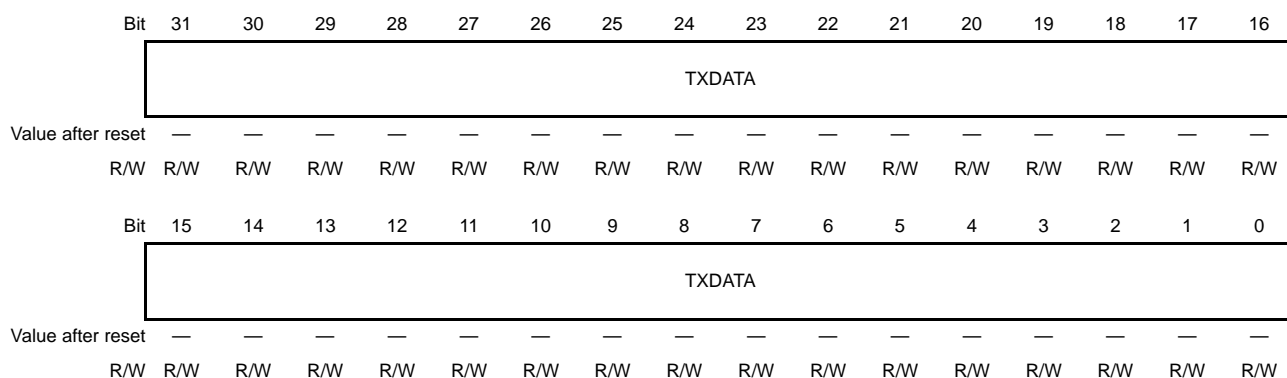


Table 20.30 PSI5nTXDATA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | TXDATA | <p>Tx Data</p> <p>Write Tx data without the Start field, Synchronization bit, and CRC field (Tx data is written in LSB first).</p> <p>Data arrangement differs depending on the transmit frame format specified in the PSI5nTXDCTRL.FRMFORMAT bit. (See Table 20.31.)</p> <p>The data is sent after these bits are written.</p> <p>Read value is always 0.</p> |

Table 20.31 PSi5nTXDATA.TXDATA Data Arrangement in Transmit Frame Format

| TXDATA Bit Position | Transmit Frame Format (PSi5nTXDCTRL.FRMFORMAT) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------------|--|----|------------------------------------|----|---------------------------------|----|----------------|----|-----------------|------|-----|------|----|-----|------|-----|----|-----|----|-----|------|-----|----|-----|------|----|------|----|-----|----|----|-----|----|----|------|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| | Frame1 "Short" | | Frame2 "Long" (4-Bit Data Nibbles) | | Frame2 "Long" (8-Bit Data Word) | | Frame3 "XLong" | | Frame4 "XXLong" | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 to 24 | All 0 | | All 0 | | All 0 | | All 0 | | All 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | Data | D19 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | D18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | Data | D7 | D17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | D6 | D16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | D5 | D15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | D4 | D14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | | | D3 | D13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | D2 | D12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | Data | D3 | Data | D7 | D11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | D2 | D6 | D10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | D1 | D5 | RAdr | X7 | D9 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | D0 | D4 | X6 | D8 | | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | RAdr | X5 | D3 | X5 | D7 | | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | X4 | D2 | X4 | D6 | | | | | | | | | | | | |
| 9 | X3 | D1 | X3 | D5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | | | | X2 | D0 | X2 | D4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | X1 | RAdr | X1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | D3 | | | | | | | | | | | |
| 6 | | | | | | | | | | | | X0 | X0 | X0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | D2 | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | FC | F2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FC | F2 | FC | F2 | FC | F2 | D1 | | | |
| 4 | | | | | | | | | | | | | | | | | F1 | F1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | F1 | F1 | F1 |
| 3 | | | | | | | | | | | | | | | | | | | F0 | F0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | SAdr | A2 | | | | | SAdr | A2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SAdr | A2 | A2 | A2 | | | | | | |
| 1 | | | | | | | | | | | | | | | | | A1 | A1 | | | | | A1 | A1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | A1 | A1 | | | | |
| 0 | | | | | | | | | | | | | | | | | | | A0 | A0 | | | | | A0 | A0 | A0 | A0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

20.3.23 PSI5nRXSPLSET — PSI5 Receive Sampling Setting Register

This register controls the sampling timing for received data.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0100_H

Value after reset: 0000 0027_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | SMPLPROD | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 20.32 PSI5nRXSPLSET Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 to 0 | SMPLPROD | Specify the value of the sampling interval counter for received data. This counter is counted up based on the psi5_com_clk signal. Specify a 16 times sampling for 1-bit length. psi5_com_clk: 80 MHz (T = 12.5 ns), Bit rate: L (125 kHz, T = 8 μs), Count cycle: 500 ns / 12.5 ns = 40 = 28 _H Setting value: 28 _H - 01 _H = 27 _H . Example settings when psi5_com_clk = 80 MHz |

| Baud Rate | Setting Value |
|-----------|-----------------|
| 125 kHz | 27 _H |
| 189 kHz | 19 _H |
| 250 kHz | 13 _H |

20.3.24 PSI5nRXSmSET — PSI5 Receive Slot m Setting Register (m = 1 to 8)

These registers set the receive slots.

Access: This register can be read/written in 32-bit units.

Address: PSI5nRXS1SET: <PSI5n_base> + 0108_H, PSI5nRXS2SET: <PSI5n_base> + 010C_H,
PSI5nRXS3SET: <PSI5n_base> + 0110_H, PSI5nRXS4SET: <PSI5n_base> + 0114_H,
PSI5nRXS5SET: <PSI5n_base> + 0118_H, PSI5nRXS6SET: <PSI5n_base> + 011C_H,
PSI5nRXS7SET: <PSI5n_base> + 0120_H, PSI5nRXS8SET: <PSI5n_base> + 0124_H

Value after reset: 00A0 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----|-----|-----|-----|------------|------------|--------|-----|-----|-----|-----------|-----|-----|-----|-----|
| | SLTEN | — | — | — | — | PAS CMP | ERR DET | LENGTH | | | | OFFSETCNT | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OFFSETCNT | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 20.33 PSI5nRXSmSET Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 | SLTEN | Slot Enable 0: Slot m is disabled. 1: Slot m is enabled. |
| 30 to 27 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 26 | PASCMP | Specifies PAS compatibility mode. 0: PSI5 mode is selected. (LSB first) 1: PAS compatibility mode is selected. (MSB first) |
| 25 | ERRDET | Specifies the error detection scheme. 0: 1-bit parity 1: 3-bit CRC |
| 24 to 20 | LENGTH | Specify the data length |
| 19 to 0 | OFFSETCNT | Specify the count value of the offset timer. When the count value reaches the specified value, the PSI5 starts to receive the data in Slot m. The value counted by psi5_com_clk is used. <ul style="list-style-type: none"> Example settings when psi5_com_clk = 80 MHz Start time: 44 μs psi5_com_clk: 80 MHz (T = 12.5 ns) Offset value: 44 us / 12.5 ns = 3520 = 0DC0_H OFFSETCNT: 0DC0_H- 0001_H = 0DBF_H |

CAUTIONS

1. In asynchronous mode (PSI5nOPMCOMM.COMMODE = 000_B), only PSI5nRXS1SET is available.
2. The ERRDET must always be set to 0 in PAS compatibility mode (PASCMP = 1).

20.3.25 PSI5nRXDATA — PSI5 Receive Data Register

This register is the receive data register.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 0128_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RXDATA | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXDATA | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.34 PSI5nRXDATA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | RXDATA | Received Data These bits hold received data that does not include a start bit and CRC/parity bits (received data is stored in LSB first). These bits are effective only when PSI5nRXMODST.RXDEXIST = 1. |

20.3.26 PSI5nRXDST — PSI5 Receive Data Status Register

This register indicates the status of received data.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 012C_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|--------|----|-----------|----|----|----|----|----|----|--------------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | RXCHKD | | RXSLOTNUM | | | | — | — | — | RXSTAT US | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.35 PSI5nRXDST Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 30 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 8 | RXCHKD | Raw Data of CRC or Parity In parity mode, setting is as below: RXCHKD[2:1]: 00 _B RXCHKD[0]: Parity value These bits are effective only when PSI5nRXMODST.RXDEXIST bit = 1 |
| 7 to 4 | RXSLOTNUM | Indicate the slot number of received PSI5nRXDATA.RXDATA. The slot number of the first received data is 1 (the slot number is from 1 to 8). These bits are effective only when PSI5nRXMODST.RXDEXIST = 1. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | RXSTATUS | Indicates the status of Rx data. 0: No error. 1: CRC, parity, or syntax error occurred. Syntax error includes the following three types: <ul style="list-style-type: none"> – An illegal start bit – Data shorter than PSI5RXSnSET.length (A long data error cannot be detected.) – Manchester code error |

20.3.27 PSI5nRXDTIM — PSI5 Receive Data IP Timer Register

This register indicates the IPTIMER value in received data.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 0130_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RXDTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXDTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.36 PSI5nRXDTIM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | RXDTIM | Specify the PSI5nIPTIMER value when the last data (CRC or Parity) is received. These bit fields are effective only when the PSI5RXMODST.RXDEXIST bit is set to 1. |

20.3.28 PSI5nRXDFIFO — PSI5 Receive Data FIFO Register

This register is the receive data FIFO register.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 0134_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RXDST | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXDST | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.37 PSI5nRXDFIFO Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 0 | RXDST | <p>These bits are used for DMA.</p> <p>These bits can be read after data is received (when PSI5nRXMODST.RXDEXIST= 1 and the INT_RXDEXIST interrupt is asserted). This register acts as FIFO whose content includes PSI5nRXDATA, PSI5nRXDST, and PSI5nRXDTIM. Read the receive data three times each. Data is read in the following order: PSI5nRXDATA, PSI5nRXDST, and then PSI5nRXDTIM.</p> <p>Exclusive access is required between PSI5nRXDFIFO and “PSI5nRXDATA, PSI5nRXDST, or PSI5nRXDTIM”.</p> |

20.3.29 PSI5nRXMODST — PSI5 Receive Module Status Register

This register is the status register for the receive module.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 0138_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|--------------------|----|----|----|----|----|----|----|--------------|
| | — | — | — | — | — | — | — | RXD SCNF ERR | — | — | — | — | — | — | — | RXD ERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | RXD FOVF | — | — | — | — | — | — | — | RXD EXIST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.38 PSI5nRXMODST Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 25 | Reserved | When read, the value after reset is read. |
| 24 | RXDSCNFERR | Indicates collision errors of synchronization pulse and reception of sensor data. 0: No error has occurred. 1: An error has occurred. When this bit is set, the INT_RXDSCNFERR interrupt is asserted. |
| 23 to 17 | Reserved | When read, the value after reset is read. |
| 16 | RXDERR | Indicates the status of receive data. 0: No error. 1: CRC, parity, or syntax error occurred. This bit is set by PSI5nRXDST.RXSTATUS bit. When this bit is set, the INT_RXDERR interrupt is asserted. |
| 15 to 9 | Reserved | When read, the value after reset is read. |
| 8 | RXDFOVF | Receive Data FIFO Overflow Flag 0: No overflow. 1: Overflow occurred. When this bit is set, the INT_RXDFOVF interrupt is asserted. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | RXDEXIST | Indicates whether or not Rx data exists. 0: No Rx data exists. 1: Rx data exists. When the PSI5nRXDATA, PSI5nRXDST, and PSI5nRXDTIM registers are read, this bit is cleared. When this bit is set, the INT_RXDEXIST interrupt is asserted. |

20.3.30 PSI5nRXMODSTCLR — PSI5 Receive Module Status Clear Register

This register is the status clear register for the receive module.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 013C_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----------------|----|----|----|----|----|----|----|------------|
| | — | — | — | — | — | — | — | RXDSCNFERR CLR | — | — | — | — | — | — | — | RXDERR CLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|-------------|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | RXDFOVF CLR | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R | R |

Table 20.39 PSI5nRXMODSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 31 to 25 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 24 | RXDSCNFERR CLR | Clears the PSI5nRXMODST.RXDSCNFERR bit. 0: No operation is done. 1: The RXDSCNFERR bit is cleared. Read value is always 0. |
| 23 to 17 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 16 | RXDERRCLR | Clears the PSI5nRXMODST.RXDERR bit. 0: No operation is done. 1: The RXDERR bit is cleared. Read value is always 0. |
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | RXDFOVFCLR | Clears the PSI5nRXMODST.RXDFOVF bit. 0: No operation is done. 1: The RXDFOVF bit is cleared. Read value is always 0. |
| 3 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

20.3.31 PSI5nRXMODSTINTEN — PSI5 Receive Module Status Interrupt Enable Register

This register is the status interrupt enable register for the receive module.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0140_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|---------------|
| | — | — | — | — | — | — | — | RXDSCNFERRINTEN | — | — | — | — | — | — | — | RXDERRINTEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | RXDFOVFINTEN | — | — | — | — | — | — | — | RXDEXISTINTEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R | R/W |

Table 20.40 PSI5nRXMODSTINTEN Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|---|
| 31 to 25 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 24 | RXDSCNFERRINTEN | INT_RXDSCNFERR Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled |
| 23 to 17 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 16 | RXDERRINTEN | INT_RXDERR Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled |
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | RXDFOVFINTEN | INT_RXDFOVF Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled |
| 3 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | RXDEXISTINTEN | INT_RXDEXIST Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled |

20.3.32 PSI5nRXMSET — PSI5 Receive Message Channel Setting Register

This register sets the channels for the receive module.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0180_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|------------|------------|------------|------------|------------|------------|------------|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | RXM8 EN | RXM7 EN | RXM6 EN | RXM5 EN | RXM4 EN | RXM3 EN | RXM2 EN | RXM1 EN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 20.41 PSI5nRXMSET Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 | RXM8EN | Controls the messaging channel for slot 8. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |
| 6 | RXM7EN | Controls the messaging channel for slot 7. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |
| 5 | RXM6EN | Controls the messaging channel for slot 6. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |
| 4 | RXM5EN | Controls the messaging channel for slot 5. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |
| 3 | RXM4EN | Controls the messaging channel for slot 4. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |
| 2 | RXM3EN | Controls the messaging channel for slot 3. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |
| 1 | RXM2EN | Controls the messaging channel for slot 2. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |
| 0 | RXM1EN | Controls the messaging channel for slot 1. 0: Messaging channel is disabled. 1: Messaging channel is enabled. |

20.3.33 PSI5nRXMRXMSG — PSI5 Receive-Message Receive Message Register

This register is the receive message register.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 0184_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|
| | CONFI GBIT | — | — | — | — | — | — | — | SERIALID | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DATAFIELD | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.42 PSI5nRXMRXMSG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 | CONFIGBIT | Configuration Bit in the Received Serial Data Frame See Figure 20.3 . |
| 30 to 24 | Reserved | When read, the value after reset is read. |
| 23 to 16 | SERIALID | Serial ID in the Received Serial Data Frame See Figure 20.3 . |
| 15 to 0 | DATAFIELD | Data Field in the Received Serial Data Frame See Figure 20.3 . |

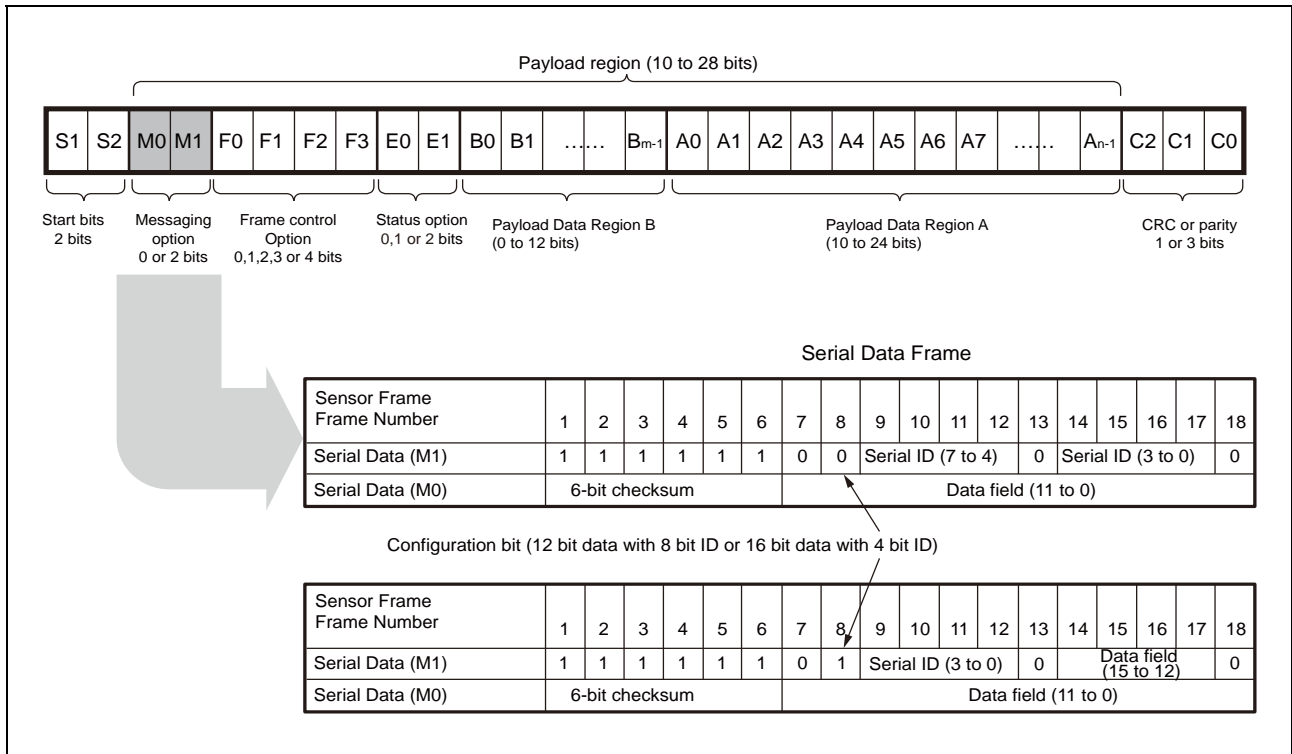


Figure 20.3 Serial Data Frame Configured by the Two Messaging Bits of the Sensor Data Frame (Messaging Channel)

20.3.34 PSI5nRXMRXST — PSI5 Receive-message Channel Receive Status Register

This register is the receive status register for the receive-message channel.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 0188_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|-------|----|----|----|----|---------|----|----|----|----|----|--------|----|----------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RXSYNC | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | RXCRC | | | | | SLOTNUM | | | | | — | — | — | RXSTATUS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.43 PSI5nRXMRXST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 30 to 19 | Reserved | When read, the value after reset is read. |
| 18 to 16 | RXSYNC | Received Synchronization Bit Raw Data (the frame numbers 7, 13, and 18 of the sensor frame) For the sensor frame, see Figure 20.3 . |
| 15, 14 | Reserved | When read, the value after reset is read. |
| 13 to 8 | RXCRC | Received CRC Raw Data |
| 7 to 4 | SLOTNUM | Received Serial Data Frame's Slot Number |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | RXSTATUS | Indicates the error state of Rx data. 0: No error. 1: CRC or Syntax error occurred. A syntax error also occurs when the receive synchronization bit has received a value other than 0 while RXSYNC = 000 _B . |

20.3.35 PSI5nRXMRXTIM — PSI5 Receive-message Channel Receive Timestamp Register

This register indicates the value of the receive-message channel receive timestamp.

Access: This register can be read in 32-bit units.

Address: <PSI5n_base> + 018C_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | RXMTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXMTIM | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.44 PSI5nRXMRXTIM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 0 | RXMTIM | Indicate the PSI5nIPTIMER value when the last data (frame number 18 of the sensor frame) is received. For the sensor frame, see Figure 20.3 . These bits are effective only when PSI5nRXMMST.RXMEXIST = 1. |

20.3.36 PSI5nRXMFIFO — PSI5 Receive-message Channel FIFO Register

This register indicates the receive-message channel FIFO information.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 0190_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | RXMFIFO | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | RXMFIFO | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.45 PSI5nRXMFIFO Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 0 | RXMFIFO | <p>These bits can be read after data is received (when PSI5nRXMMST.RXMEXIST= 1: INT_RXMEXIST interrupt is asserted). This register acts as FIFO whose content includes PSI5nRXMRXMSG, PSI5nRXMRXST, and PSI5nRXMTIM. Read the received data three times each.</p> <p>Data is read in the following order: PSI5nRXMRXMSG, PSI5nRXMRXST, and then PSI5nRXMTIM.</p> <p>Exclusive access is required between PSI5nRXMFIFO and "PSI5nRXMRXMSG, PSI5nRXMRXST, or PSI5nRXMTIM".</p> |

20.3.37 PSI5nRXMMST — PSI5 Receive-message Channel Module Status Register

This register is the status register for the receive-message channel module.

Access: This register can only be read in 32-bit units.

Address: <PSI5n_base> + 0194_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|-------------|----|----|----|----|----|----|----|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RXM ERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | RXM FOVF | — | — | — | — | — | — | — | RXM EXIST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 20.46 PSI5nRXMMST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 25 | Reserved | When read, the value after reset is read. |
| 16 | RXMERR | Rx Message Error 0: No error. 1: Error occurred. When this bit is set, the INT_RXMERR interrupt is asserted. |
| 15 to 9 | Reserved | When read, the value after reset is read. |
| 8 | RXMFOVF | Rx Message FIFO Overflow Flag 0: No overflow. 1: Overflow occurred. When this bit is set, the INT_RXMFOVF interrupt is asserted. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | RXMEXIST | Indicates whether Rx Serial Data Frame exists. 0: No Rx serial data exists. 1: Rx serial data exists. When the PSI5nRXMRXMSG, PSI5nRXMRXST, and PSI5nRXMTIM registers are read, this bit is cleared. When this bit is set, the INT_RXMEXIST interrupt is asserted. |

20.3.38 PSI5nRXMMSTCLR— PSI5 Receive-message Channel Module Status Clear Register

This register is used to clear the receive-message channel module status register.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 0198_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|-------------|----|----|----|----|----|----|----|------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RXMERR CLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | RXMFOVF CLR | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R | R |

Table 20.47 PSI5nRXMMSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 25 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 16 | RXMERRCLR | Clears the status of the PSI5nRXMMST.RXMERR bit. 0: No operation is done. 1: The RXMERR bit is cleared. Read value is always 0. |
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | RXMFOVFCLR | Clears the status of the PSI5nRXMMST.RXMFOVF bit. 0: No operation is done. 1: The RXMFOVF bit is cleared. Read value is always 0. |
| 3 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

20.3.39 PSI5nRXMMSTINTEN — PSI5 Receive-message Channel Module Status Interrupt Enable Register

This register controls interrupts from the receive-message channel module.

Access: This register can be read/written in 32-bit units.

Address: <PSI5n_base> + 019C_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----------------------|----|----|----|----|----|----|----|-----------------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RXM ERR INTEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | RXM FOVF INTEN | — | — | — | — | — | — | — | RXM EXIST INTEN |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R | R | R/W |

Table 20.48 PSI5nRXMMSTINTEN Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|---|
| 31 to 25 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 16 | RXMERR INTEN | INT_RXMERR Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled |
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | RXMFOVF INTEN | INT_RXMFOVF Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled |
| 3 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | RXMEXIST INTEN | INT_RXMEXIST Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled |

20.3.40 PSI5TSSEL — PSI5 Timestamp Function Mode Selection Register

This register specifies the timestamp setting.

This register can be controlled when the PSI5 is stopped (PSI5nCHCTRL.CHEN = 0).

Access: This register can be read/written in 32-bit units.

Address: <PSI50_base> + 3000_H

Value after reset: 0000 0000_H This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PSI5MS SEL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 20.49 PSI5TSSEL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PSI5MSSEL0 | Timestamp Mode Selection 0: Inputs the timestamp count-up timing and clearing signals from PSI51 to PSI50. 1: Inputs the timestamp count-up timing and clearing signals from PSI50 to PSI51. |

20.4 Interrupt

The PSI5 has 10 interrupt signals.

Table 20.50 PSI5 Interrupt Signals and Corresponding Registers

| Interrupt Signal | Function | Source Register | Interrupt Enable Register |
|------------------|---|-----------------------------|-------------------------------------|
| | | Bit | Bit |
| INT_SYNCED | Synchronization end interrupt | PSI5nTXST. SYNCED | PSI5nTXSTINTEN. SYNCEDINTEN |
| INT_SYNCST | Synchronization start interrupt | PSI5nTXST. SYNCST | PSI5nTXSTINTEN. SYNCSTINTEN |
| INT_TXEMPTY | Transmit data empty interrupt | PSI5nTXST. TXEMPTY | PSI5nTXSTINTEN. TXEMPTYINTEN |
| INT_RXDSCNFERR | Synchronization pulse and receive data conflict error interrupt | PSI5nRXMODST. RXDSCNFERR | PSI5nRXMODSTINTEN. RXDSCNFINTEN |
| INT_RXDERR | Receive data error interrupt | PSI5nRXMODST. RXDERR | PSI5nRXMODSTINTEN. RXDERRINTEN |
| INT_RXDFOVF | Receive data FIFO overflow interrupt | PSI5nRXMODST. RXDFOVF | PSI5nRXMODSTINTEN. RXDFOVFINTEN |
| INT_RXDEXIST | Receive data exist interrupt | PSI5nRXMODST. RXDEXIST | PSI5nRXMODSTINTEN. RXDEXISTINTEN |
| INT_RXMFOVF | Receive message FIFO overflow interrupt | PSI5nRXMMST. RXMFOVF | PSI5nRXMMSTINTEN. RXMFOVFINTEN |
| INT_RXMERR | Receive message error interrupt | PSI5nRXMMST. RXMERR | PSI5nRXMMSTINTEN. RXMERRINTEN |
| INT_RXMEXIST | Receive message exist interrupt | PSI5nRXMMST. RXMEXIST | PSI5nRXMMSTINTEN. RXMEXISTINTEN |

The following figure shows the relationship between the PSI5 interrupt signals, and RH850/P1M-E's interrupt controller INTC and DMA.

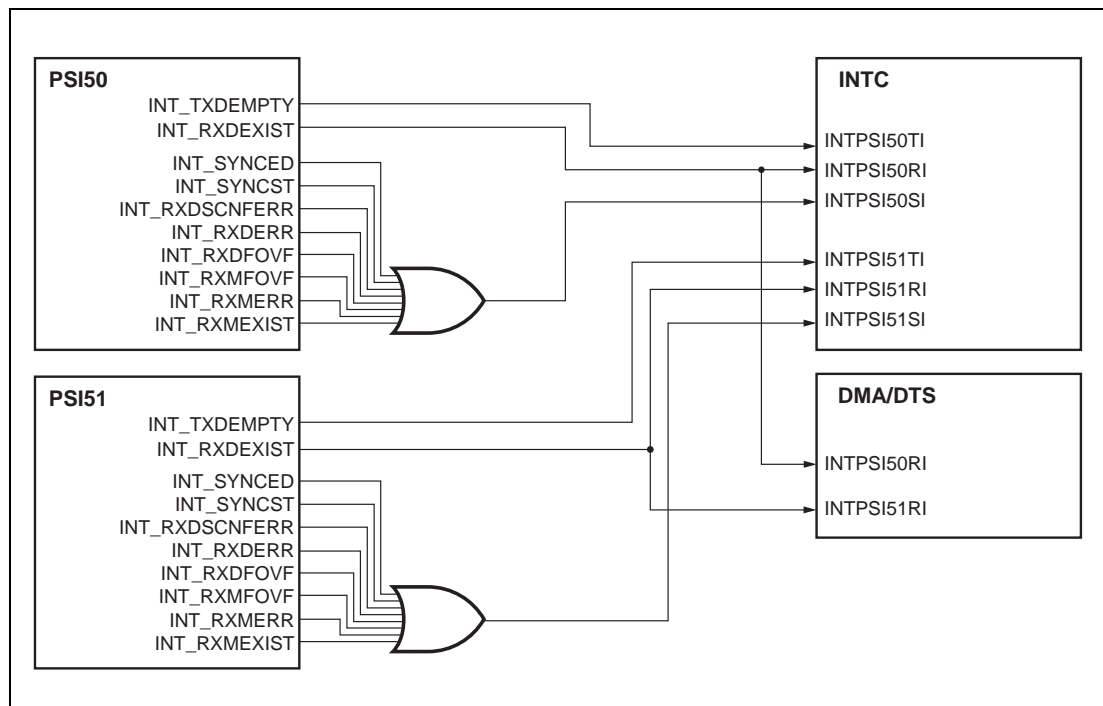


Figure 20.4 PSI5 Interrupt Signals

20.5 Operation

20.5.1 Setting Operation Mode

Before the PSI5n starts operating, initial settings including the sensor connection status and operation mode must be configured. The following describes necessary initial settings.

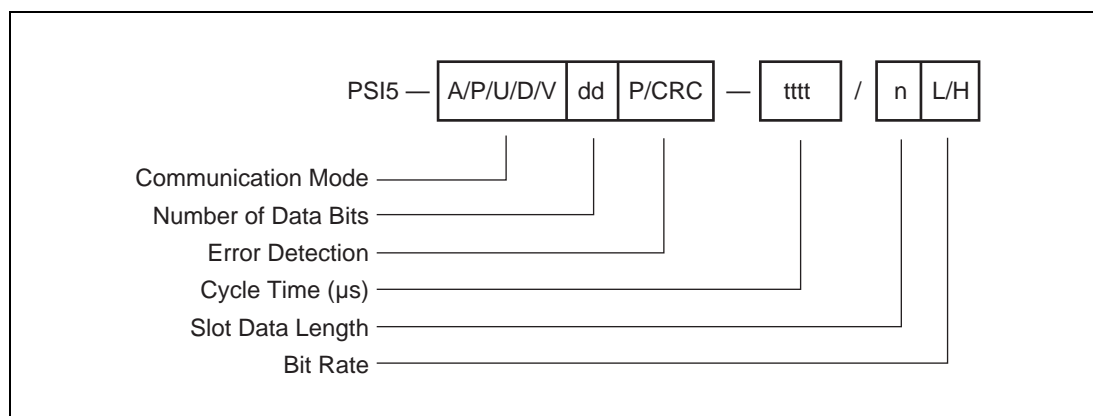


Figure 20.5 Setting PSI5 Operation Mode

Table 20.51 Setting Operation Mode

| Item | | Description |
|----------------------|------|---|
| Communication Modes | A | Asynchronous Mode |
| | P | Synchronous Parallel Bus Mode |
| | U | Synchronous Universal Bus Mode |
| | D | Synchronous Daisy Chain Bus Mode |
| | V | Variable Time Triggered Synchronous Operation Mode |
| Number of data bits | dd | Data length of each slot specified in the PSI5nRXSmSET.LENGTH bit (10 to 28 bits) |
| Error Detection | P | One Parity Bit |
| | CRC | Three Bits CRC |
| Cycle time | tttt | Cycle Time (Minimum cycle time allowed in variable time triggered synchronous operation mode)(µs) |
| Slot count/cycle (n) | n | Select the slot to be used in the PSI5nRXMSET. |
| Bit Rate | L | 125 Kbps |
| | H | 189 Kbps |

20.5.1.1 Setting for Transmit/Receive Mode

Set PSI5nOPMCOMM, PSI5nOPMBITRATE, and PSI5nOPMCYCT to the value corresponding to the specification of the PS15 sensor and its connection status.

20.5.1.2 Setting for Data Reception

The RH850/P1M-E receives data on psi5_com_clk.

A 16-times sampling is used for bit detection.

The bit length depends on the setting of PSI5nRXSPLSET.SMPLPROD[7:0].

- PSI5nRXSPLSET.SMPLPROD[7:0]: Set the sampling interval.

Set the value counted by psi5_com_clk. The RH850/P1M-E uses a 16-times sampling.

20.5.1.3 Initial Setting for Asynchronous Mode

In asynchronous mode, only PSI5nRXS1SET is available. Writing 1 to SLTEN in PSI5nRXS2SET to PSI5nRXS8SET is prohibited (so disable them).

In asynchronous mode, set the SLTEN, PASCMP, ERRDET, and LENGTH fields in PSI5nRXS1SET. The OFFSETCNT field is not available.

20.5.1.4 Initial Setting for Synchronous Mode

In synchronous mode, specify all fields of PSI5nRXS1SET to PSI5nRXS8SET appropriately.

The recommended setting values for PSI5nRXS1SET.OFFSETCNT and PSI5nRXS2SET to PSI5nRXS8SET.OFFSETCNT can be obtained as the results of the formula in **Figure 20.6** and **Figure 20.7**. In these figures, tcom_clk indicates the time for one psi5_com_clk cycle. Other variables are the variables specified by the PSI5 standard.

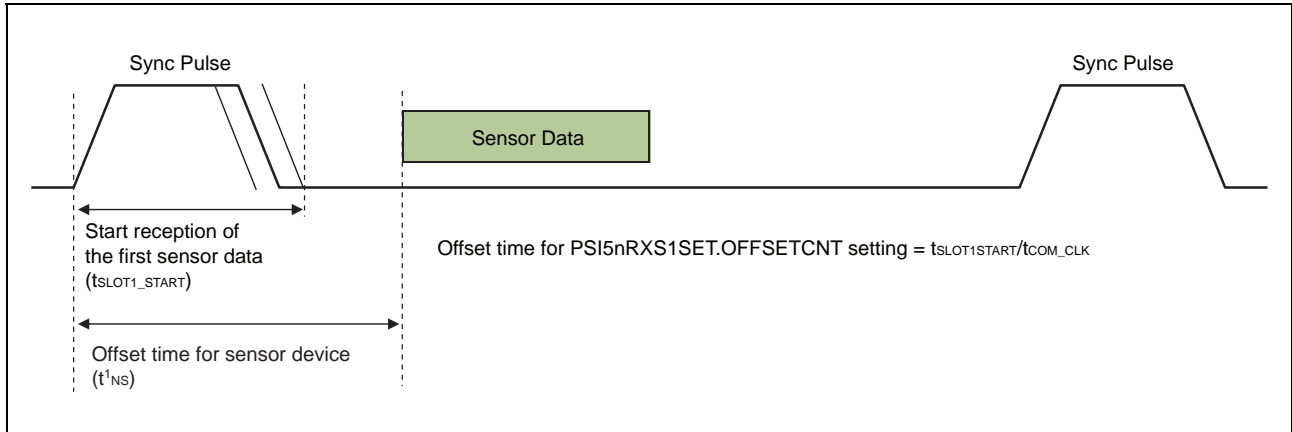


Figure 20.6 Recommended Setting of PSI5nRXS1SET.OFFSETCNT

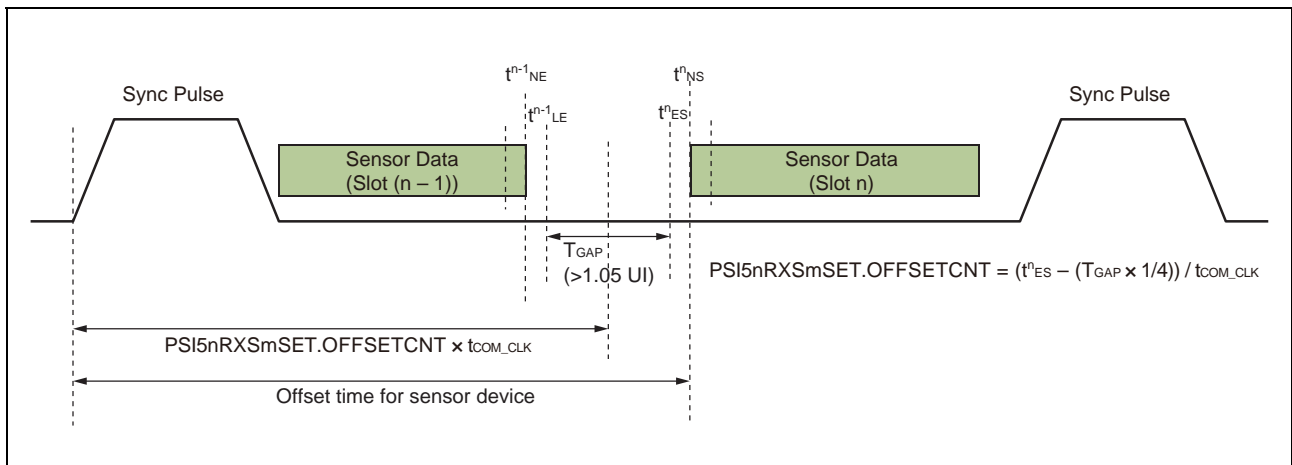


Figure 20.7 Recommended Setting of PSI5nRXSmSET.OFFSETCNT (m = 2 to 8)

20.5.1.5 Setting for Serial Message Reception

To receive a serial message, write 1 to PSI5nRXMSET. RXMmEN (m = 1 to 8) in the corresponding slot.

20.5.1.6 Initial Setting in Asynchronous Mode

In asynchronous mode, registers for transmission are disabled except PSI5nTXSETTING.DEFTXVAL fields. Use the value after reset.

The PSI5 outputs the value set in PSI5nTXSETTING.DEFTXVAL after the start of operation (although the standard does not define output data in asynchronous mode). The value after reset is 1.

20.5.1.7 Initial Setting in Synchronous Mode

Set the value in PSI5nTXSETTING.

PSI5nTXSETTING.DEFTXVAL sets a synchronization pulse value for the period during which data is not transmitted. In Tooth GAP mode, set PSI5nTXSETTING.DEFTXVAL to 1. In pulse width mode, the value is not specified in the standard. Set the value specified by a system designer.

In variable time triggered synchronous operation mode, the user must specify the timing for synchronization pulse output.

When issuing a synchronization pulse, write 1 to the PSI5nSYNCCTRL.VALTIMSYNC bits.

Set the data format to be transmitted to the sensor in PSI5nTXDCTRL.FRMFORMAT.

20.5.1.8 Setting for Timestamping

A timestamp is appended to the received data so that the order of the received data and messages can be identified.

Timestamping is available in two modes: master mode for generation of the timing for counting up within the PSI5, and slave mode for counting with an input signal from the master. In master mode, the count-up timing is output to the output signal psi5_ts_tick_out. In slave mode, the input signal psi5_ts_tick_in is used for counting up.

To use a timestamp, set PSI5nIPTIMERCTRL.IPTIMEN to 1. Specify master or slave in PSI5nIPTIMERCTRL.MSTSLV.

In master mode, the count-up timing is complete when the internal baud rate counter is complete. The value to complete the baud rate counter is the setting of PSI5nOPMBITRATE.BITRATECNT.

The timestamp can be cleared by writing 1 to PSI5nIPTIMERCTRL.TSCLR irrespective of master or slave. In master mode, the clearing timing is output to the output signal psi5_ts_clr_out. In slave mode, the input signal psi5_ts_clr_in from the master can be used to clear the timestamp.

(1) Setting of Master/Slave Mode in Timestamp Mode

For PSI50 and PSI51, the timestamp mode can be selected from master mode and slave mode.

The following shows the setting of master or slave mode.

Table 20.52 Setting of Master/Slave Mode

| PSI5TSSEL. PSI5MSSEL0 Bit | IPTIMERCTRL. MSTSLV Bit in PSI50 | IPTIMERCTRL. MSTSLV Bit in PSI51 | Timestamp Mode | | Operation |
|---------------------------------|--|--|----------------------|----------------------|---|
| | | | PSI50 | PSI51 | |
| 1 | 1 | 0 | Master of PSI51 | Slave of PSI50 | PSI50 operates as master and PSI51 operates as slave. |
| 1 | 1 | 1 | Master (No slave) | Master (No slave) | PSI50 and PSI51 operate in master mode without slaves. |
| 0 | 0 | 1 | Slave of PSI51 | Master of PSI50 | PSI51 operates as master, and PSI50 operates as slave. |
| 0 | 1 | 1 | Master (No slave) | Master (No slave) | PSI50 and PSI51 operates in master mode without slaves. |
| Other than above | | | Setting prohibited | | |

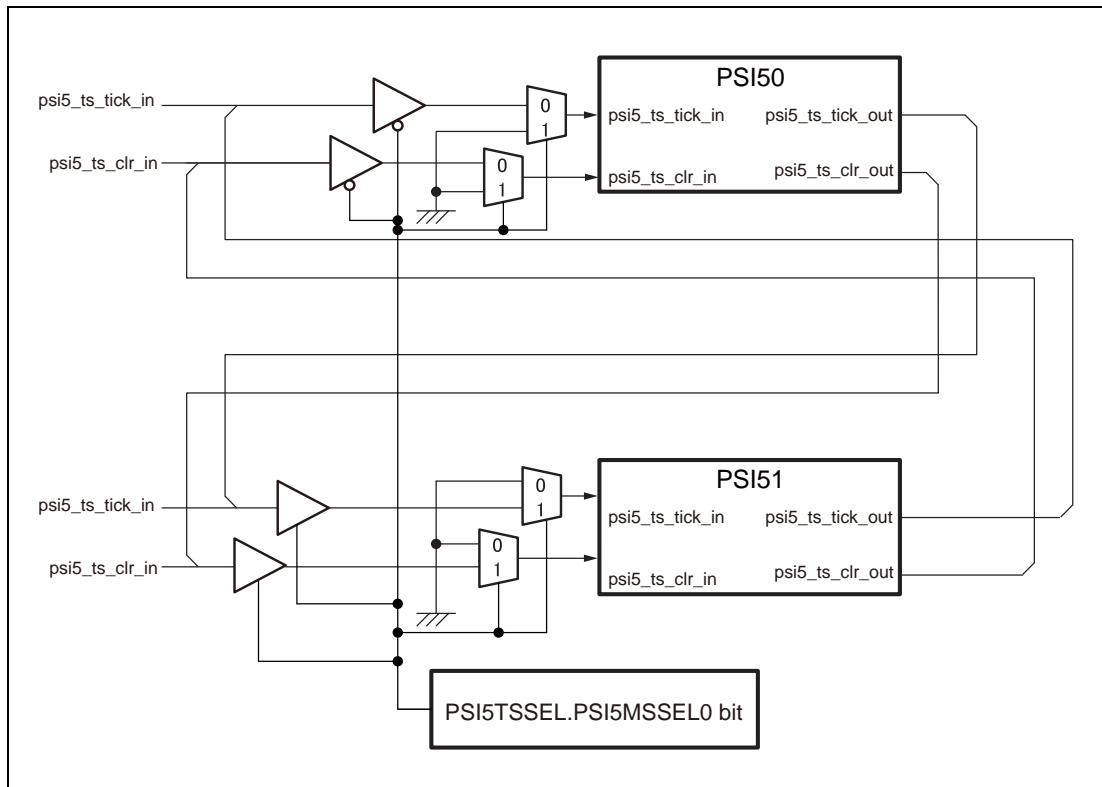


Figure 20.8 Relationship Between PSI50 and PSI51

20.5.2 Operation Flow

20.5.2.1 Starting Operation

Figure 20.9 shows the flow from initialization to the start of operation. When resetting the sensor or releasing it from reset, issue instructions to the PHY as these operations are performed by the PHY.

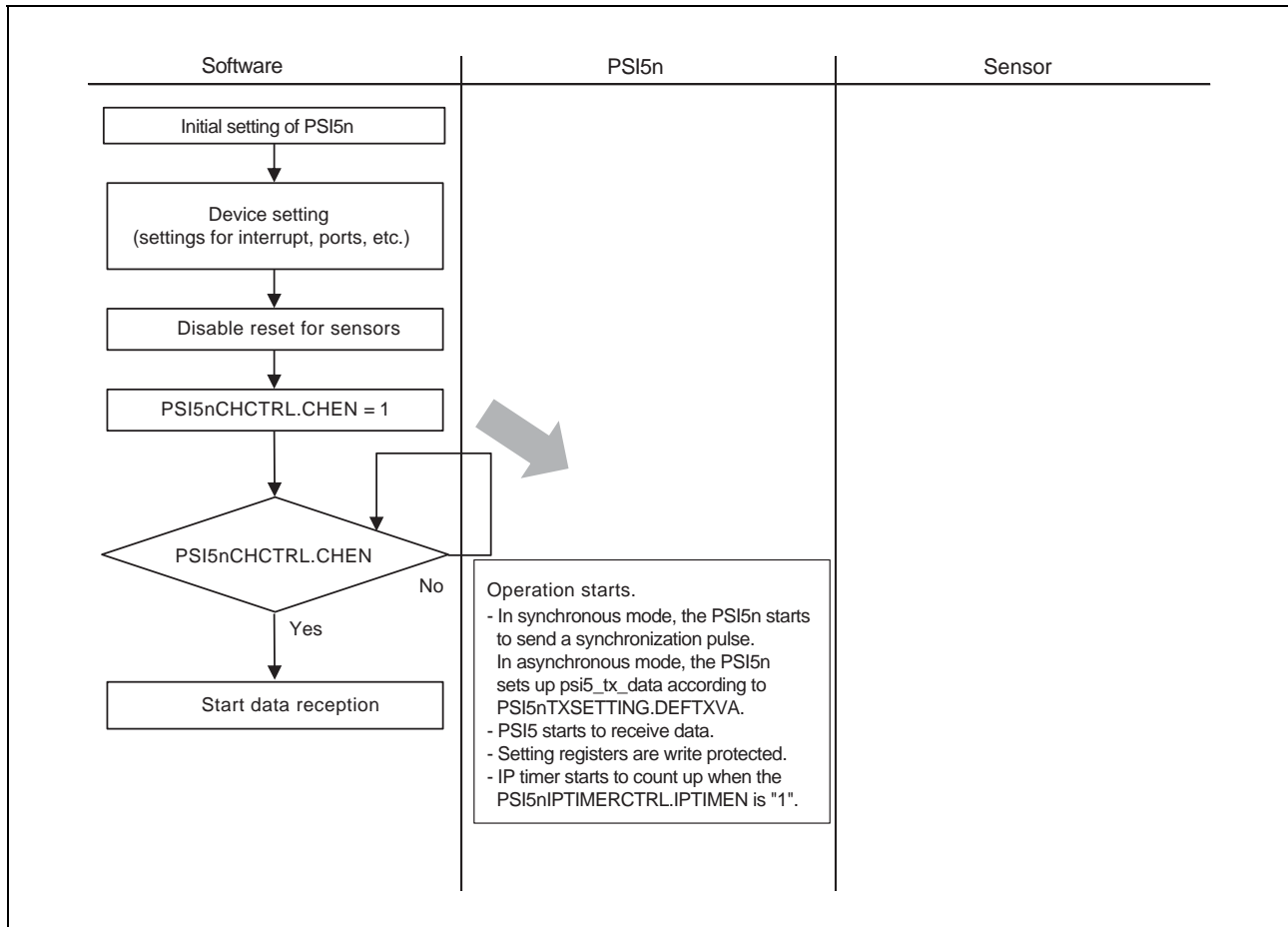


Figure 20.9 Flow of Starting Operation

20.5.2.2 Flow of Data Reception

Figure 20.10 shows the flow of data reception. The interrupt enable (PSI5nRXMODSTINTEN.RXDEXISTINTEN = 1) in the flowchart is not required when polling is used for detection of reception, or when it is always enabled (this applies to enabling of other interrupts in the subsequent sections).

The received data without the start bit and CRC/parity bits can be read out from PSI5nRXDATA.RXDATA.

Regarding the FIFO buffer shown in **Figure 20.10**, when data are received, that state is retained, and even if the register is empty, the received data are set from the FIFO buffer to the register. In cases of overflow because of delays in the reading out of data, new data are written to the buffer.

When reading data from the FIFO buffer, check that the value of PSI5nRXMDST.RXDEXIST is 0. Since PSI5nRXMDST.RXDEXIST being 1 indicates that data which have not been read out remain in the FIFO buffer, repeat the processing for reading.

The PSI5nRXDST.RXSTATUS is set to 1 in response to a start-bit error, Manchester code error, CRC error, or parity error during data reception. Specifically, in cases where multiple errors are detected during the reception of a single frame, the FIFO buffer for received data is updated every time an error occurs.

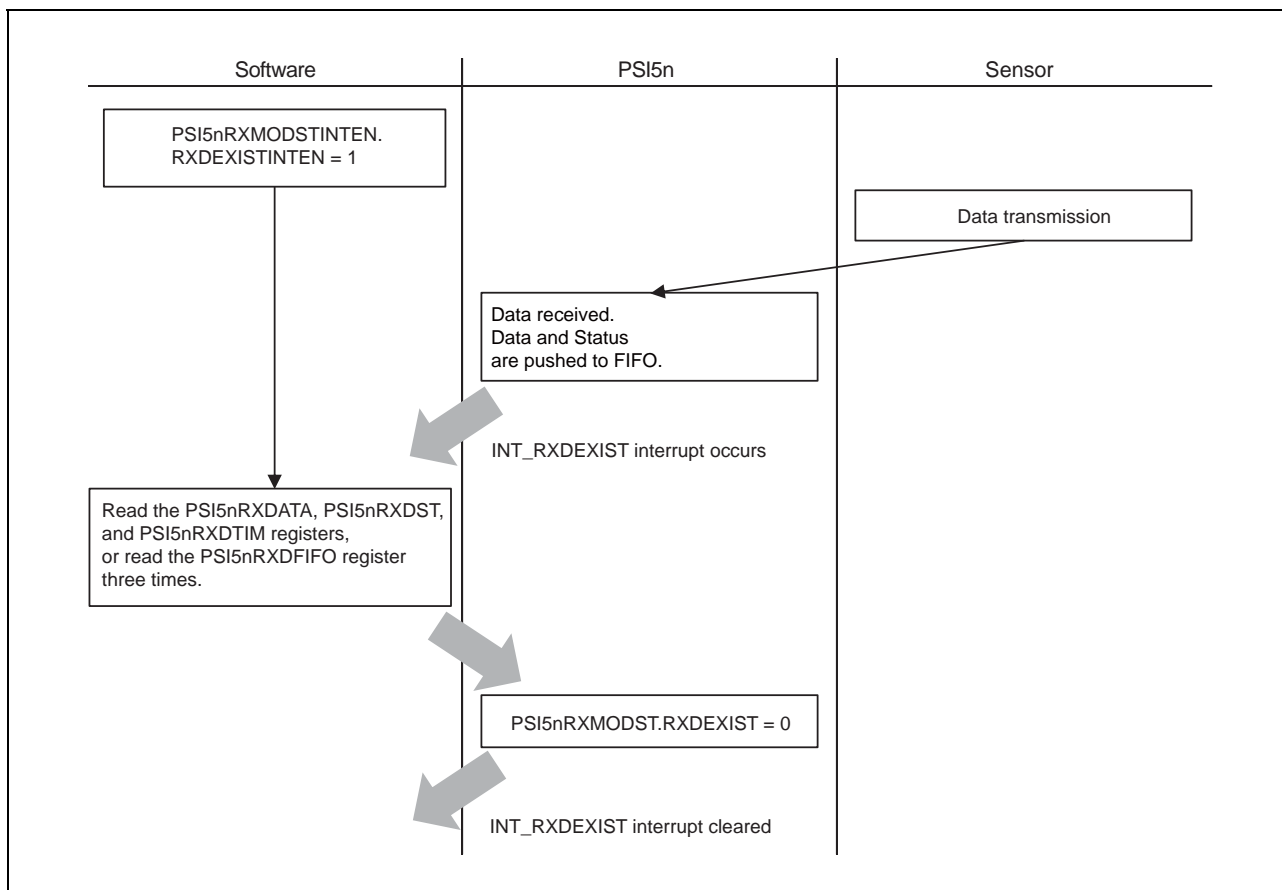


Figure 20.10 Data Reception Flow

20.5.2.3 Flow of Data Transmission

Figure 20.11 shows the flow of data transmission.

Transmit data arrangement to be written to the PSI5nTXDATA.TXDATA differs depending on the transmit frame format specified in the PSI5nTXDCTRL.FRMFORMAT bit (See **Table 20.31**). The data must not include a start bit, synchronization bit, or CRC (these are automatically appended).

Once PSI5nTXDCTRL.FRMFORMAT has been set, it does not need to be set again after that.

The PSI5 module does not have a FIFO buffer for transmission, so processing for transmission is sequential. When you intend to write to the transmission buffer, read the state of the buffer empty flag to check if it currently indicates the buffer empty state (PSI5nTXST.TXDEEMPTY = 1). If the transmission buffer holds data to be transmitted at the time of writing (PSI5nTXST.TXDEEMPTY = 0), transmitted data become undefined when data are written to the transmission buffer.

PSI5nTXST.TXDEEMPTY becomes 1 to indicate the completion of transmission when reading of the last bit of the data from the transmission buffer starts.

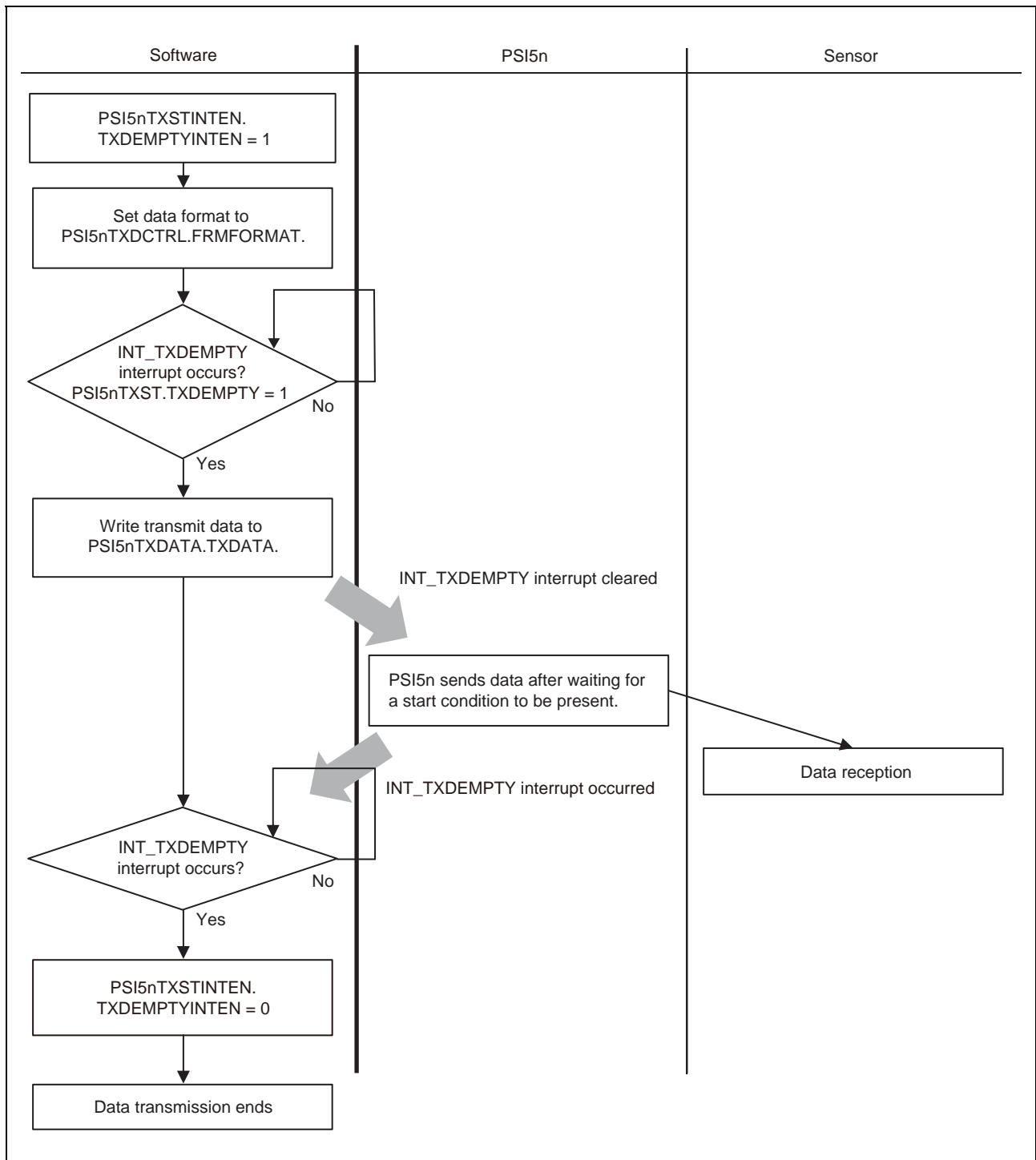


Figure 20.11 Data Transmission Flow

20.5.2.4 Serial Message Reception Flow

Figure 20.12 shows the flow of serial message reception.

The received serial message can be read out from PSI5nRXMRXMSG.RXDATA.

In Figure 20.12, the FIFO buffer holds the serial message and its state and the received data are set from the FIFO buffer to the register if the register is empty. In cases of overflow because of delays in the reading out of data, new data are written to the buffer.

When reading data from the FIFO buffer, check that the value of PSI5nRXMMST.RXMEXIST is 0. Since PSI5nRXMMST.RXMEXIST being 1 indicates that data which have not been read out remain in the FIFO buffer, repeat the processing for reading.

If either a CRC error or a receive data error occurs after the detection of the start bit in the message (after reception of M1 (Frame No.) = 11111_B), PSI5nRXMRXST.RXSTATUS is set to 1. In this case, handling of the received message is system-dependent. When an receive data error occurs, the message up to that data is stored in order from the least-significant bit of PSI5nRXMRXMSG.

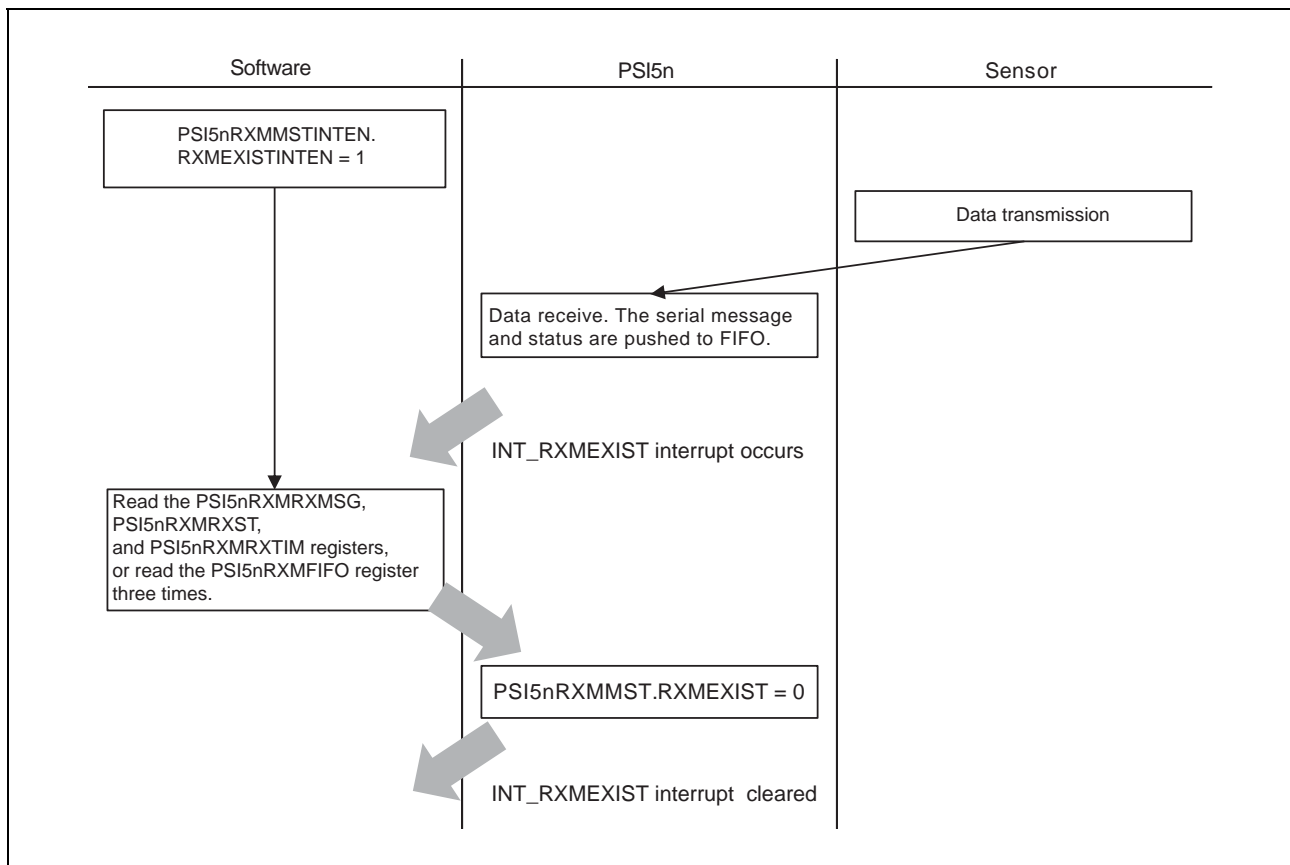


Figure 20.12 Serial Message Reception Flow

20.5.2.5 Parity and CRC Errors in Received Data

Error flags (PSI5nRXMODST.RXDERR and PSI5nRXMMST.RXMERR) are set in response to the detection of parity or CRC errors in received data. Reception operations proceed whether or not errors are detected.

Each of the error flags has a corresponding clearing bit (PSI5nRXMODSTCLR.RXDERRCLR and PSI5nRXMMSTCLR.RXMERRCLR).

20.5.2.5.1 Operation after Error Detection in Asynchronous Mode and PAS Compatibility Mode

When external noise and so on lead to errors in the Manchester code during the reception of data while the PSI5 is in the asynchronous mode or PAS compatibility mode, reception of the data being received is assumed to be complete, the parity or CRC error flag is set, and the data are stored in the reception FIFO buffer. Even when a Manchester code error occurs, the next data reception starts immediately. Due to this specification, even if a Manchester code error occurs for one bit because of external noise during data reception, the following data bit is assumed to be the start bit, and reception starts. In this situation, multiple Manchester code errors may occur and multiple receptioncomplete interrupts can be generated. In such cases, the maximum number of errors corresponds to the number of bits in the unit of data.

Accordingly, this raises the possibility of producing an overflow. If an overflow occurs even though the data retrieval assumed during system designing is performed, the transfer may be malfunctioning due to factors such as noise.

In synchronous mode, data reception is halted at slot $n + 1$ in response to an error in the Manchester code while reception in slot n is in progress. For this reason, even if further errors occur or data is resent while in slot n , they are ignored.

20.5.3 PAS Compatibility Mode

The RH850/P1M-E has a PAS compatibility mode. The PAS compatibility mode supported by the RH850/P1M-E is as shown in **Table 20.7**.

In PAS compatibility mode, set 1 in PSI5nRXS1SET.PASCMP. At that time, set 0 (parity) in PSI5nRXS1SET.ERRDET.

20.5.4 Baud Rate

The communication clock is generated, by dividing 1 cycle of psi5_com_clk by 1 to 2¹⁶.

$$\begin{aligned} 1 \text{ bit cycle waveform} &= (\text{value set in PSI5nOPMBITRATE.BITRATECNT}) / \text{psi5_com_clk} \\ \text{Baud rate} &= 1/1 \text{ bit cycle waveform} \end{aligned}$$

The following shows an example baud rate setting.

Table 20.53 Example Baud Rate Setting

| Baud Rate [Kbps] | psi5_com_clk | | Value Set in PSI5nOPMBITRATE.BITRATECNT | Bit Time [μs] |
|------------------|-----------------|------------|---|---------------|
| | Frequency [MHz] | Cycle [ns] | | |
| 125 | 80 | 12.5 | 29F _H | 8.4 |
| 189 | 80 | 12.5 | 1BC _H | 5.57 |
| 250 | 80 | 12.5 | 14F _H | 4.2 |

Section 21 Window Watchdog Timer A (WDTA)

The purpose of the Window Watchdog Timer A (WDTA) is to detect deadlock of CPU operation. WDTA is implemented 1 unit. This timer generates an error signal if the counter is not cleared by CPU within a certain counter value. If the timer is cleared, MCU can keep normal operation.

21.1 Features of RH850/P1M-E WDTA

21.1.1 Units and Channels

This microcontroller has the following number of WDTA units.

Table 21.1 Number of Units

| Product | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 1 | 1 |
| Name | WDTA0 | WDTA0 |

Table 21.2 Unit Configurations and Channels

| Unit Name WDTAn | Channels per Unit | RH850/P1M-E 100 pins (1 ch) | RH850/P1M-E 144 pins (1 ch) |
|--------------------|----------------------|-----------------------------------|-----------------------------------|
| WDTA0 | 1 | √ | √ |

Note: The channel names are same as those of the corresponding units.

Table 21.3 Index

| Index | Description |
|-------|--|
| n | Throughout this section, the individual WDTA units are identified by the index “n” (n = 0): for example, WDTAnWDTE (n = 0) is the WDTAn enable register. |

21.1.2 Register Base Address

WDTAn base addresses are listed in the following table.

WDTAn register addresses are given as offsets from the base addresses throughout the section.

Table 21.4 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <WDTA0_base> | FFD7 4000 _H |

21.1.3 Clock Supply

Clock supply by and to WDTAn is listed in the following table.

Table 21.5 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|--|
| WDTA0 | WDTATCKI | High-speed mode: CLK_IOSC: 8 MHz Low-speed mode: CLK_IOSC/32: 250 kHz |
| | PCLK | High-speed peripheral clock CLK_HSB |

21.1.4 Interrupt Request

WDTAn interrupt requests are listed in the following table.

Table 21.6 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|---------------------|------------------|------------------------|
| WDTA0 | | | |
| INTWDTA0 | WDTA0 75% interrupt | 9 | — |

21.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

21.1.6 External Input/Output Signals

WDTAn has no external pin.

21.2 Overview

21.2.1 Functional Overview

The WDTA has the following functions:

- Selection of the operation mode after release from the reset state, by using start-up options

Starting/stopping of the counter after reset, the setting of the overflow time of the counter, enabling/disabling of the VAC function, and clock mode can be selected. WDTA startup options are described in **Table 21.7, WDTA Start-up Options**.

- WDTA trigger function

Writing an activation code to the WDTA trigger register starts WDTA and restarts the counter. Activation codes include fixed activation codes and variable activation codes (VAC function). In a variable activation code, a different value from the previous time (variable value) is written to the WDTA trigger register, which causes the counter to be restarted.

- 75% interrupt request signals

An interrupt request signal can be generated when the WDTA counter reaches 75% of the overflow interval time (this function can be enabled or disabled by the setting of WDTAnMD.WDTAnWIE).

- Window function

The period during which writing to the WDTA trigger register is valid (window-open period) can be set. Writing to the WDTA trigger register at a time outside the window-open period causes an error.

- WDTA error detection function

When an error is detected, an error signal WDTAnTERR is generated. The error signal WDTAnTERR is connected to the ECM module.

For details about the error sources, refer to **Section 21.5.3, WDTA Error Detection**.

Table 21.7 WDTA Start-up Options

| Start-up Option | Function | Description | Option Byte |
|-----------------|--|---|--------------------|
| OPWDRUN | Start mode setting | Specifies the start mode. 0: Software trigger start mode 1: Default start mode For details, refer to Section 21.5.1, WDTA after Release from the Reset State. | OPBT0.OPBT0[31] |
| OPWDOVF[2:0] | Overflow interval time reset value setting | Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0]. | OPBT0.OPBT0[27:25] |
| OPWDVAC | Variable Activation Code (VAC) selection | Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) When WDTAnWDTE is selected, the value to be written to the register (activation code) is fixed (AC_H). When WDTAnEVAC is selected, the activation code to be written to the register is variable. For details, refer to Section 21.5.2, WDTA Trigger, Section 21.5.2.1, Calculating an Activation Code when the VAC Function is Used. | OPBT0.OPBT0[22] |
| OPWDMDS | Clock mode selection | Selects the counter clock source of WDTAn. 0: High-speed mode (high-speed internal oscillation: 8 MHz) 1: Low-speed mode (high-speed internal oscillation/32: 250 kHz) | OPBT0.OPBT0[21] |

21.2.2 Block Diagram

Figure 21.1 shows the main components of the WDTA.

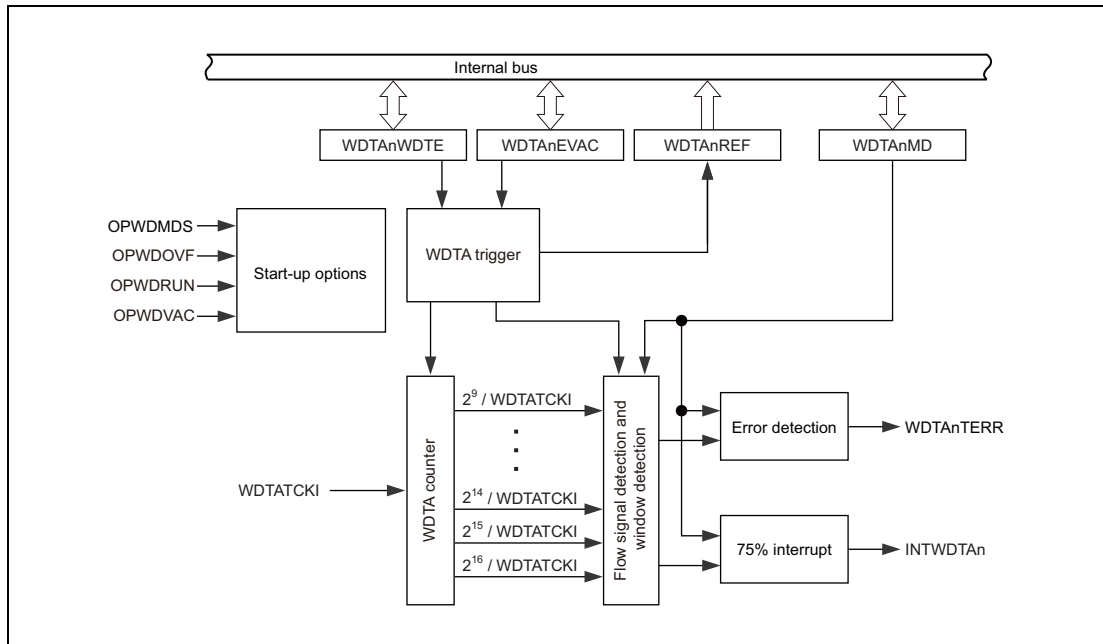


Figure 21.1 Block Diagram of the WDTA

21.3 Registers

21.3.1 List of Registers

WDTA registers are listed in the following table.

For details about <WDTAn_base>, see **Section 21.1.2, Register Base Address**.

Table 21.8 List of Registers

| Module Name | Register Name | Symbol | Address |
|-------------|-------------------------------|-----------|----------------------------------|
| WDTAn | WDTA enable register | WDTAnWDTE | <WDTAn_base> + 0000 _H |
| WDTAn | WDTA enable VAC register | WDTAnEVAC | <WDTAn_base> + 0004 _H |
| WDTAn | WDTA reference value register | WDTAnREF | <WDTAn_base> + 0008 _H |
| WDTAn | WDTA mode register | WDTAnMD | <WDTAn_base> + 000C _H |

21.3.2 WDTAnWDTE — WDTA Enable Register

This register is the WDTA trigger register if the VAC function is not used (start-up option OPWDVAC = 0).

Writing AC_H to this register generates a WDTA trigger and starts or restarts the WDTA counter. Refer to **Section 21.5.2, WDTA Trigger**, for details.

The behavior of this register depends on the setting of the start-up option (OPWDVAC). Refer to **Table 21.11, WDTAnWDTE Behavior**.

Access: This register can be read/written in 8-bit units.

Address: <WDTAn_base> + 0000_H

Value after reset: The initial value depends on the start-up options OPWDRUN and OPWDVAC. Refer to **Table 21.10, WDTAnRUN Value after a Reset**. This register is initialized by a reset of any type.

| | | | | | | | | |
|-------------------|---------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WDTAnRUN[7:0] | | | | | | | |
| Value after reset | | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 21.9 WDTAnWDTE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 7 to 0 | WDTAnRUN [7:0] | Writing a fixed activation code (AC _H) generates a WDTA trigger to control WDTAn count start and restart. Writing a value other than AC _H generates an error. After WDTAn starts, it cannot be stopped. When these bits are read or written, see Table 21.11, WDTAnWDTE Behavior . |

The WDTAnRUN7 bit is only valid if the VAC function is disabled (OPWDVAC = 0). **Table 21.10** lists the values of the WDTAnRUN7 bit after reset according to the start-up options.

Table 21.10 WDTAnRUN Value after a Reset

| Start-up options | | Start Mode | Value of WDTAnRUN7 after Reset |
|------------------|---------|------------------------|--------------------------------|
| OPWDVAC | OPWDRUN | | |
| 0 | 1 | Default start | 1 |
| | 0 | Software trigger start | 0 |

The behavior of WDTAnWDTE during read/write accesses depends on the OPWDVAC setting, as shown in **Table 21.12, WDTAnWDTE Behavior**.

Table 21.11 WDTAnWDTE Behavior

| OPWDVAC | Description | WDTAnWDTE | |
|---------|---|--|--|
| | | Read | Write |
| 0 | The VAC function is disabled. WDTAnWDTE is enabled. | 2C _H is read (before WDTAn starts in software trigger start mode). AC _H is read (after WDTAn has started). | WDTA trigger Write AC _H ^{*1} |
| 1 | The VAC function is enabled. WDTAnWDTE is disabled. | 2C _H is read. | Writing is ignored. |

Note 1. Any other write value will cause an error.

21.3.3 WDTAnEVAC — WDTA Enable VAC Register

This register is the WDTA trigger register when the VAC function is used (start-up option OPWDVAC = 1).

Writing a correct activation code to this register generates a WDTA trigger and starts or restarts the WDTA counter. For details, see **Section 21.5.2, WDTA Trigger**. For details about the activation codes when the VAC function is used, see **Section 21.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

The behavior of this register depends on the setting of the start-up option (OPWDVAC). Refer to **Table 21.14, WDTAnEVAC Behavior**.

Access: This register can be read/written in 8-bit units.

Address: <WDTAn_base> + 0004_H

Value after reset: The initial value depends on the start-up options (OPWDRUN and OPWDVAC). Refer to **Table 21.13, WDTAnEVAC7 Value after a Reset**. This register is initialized by a reset of any type.

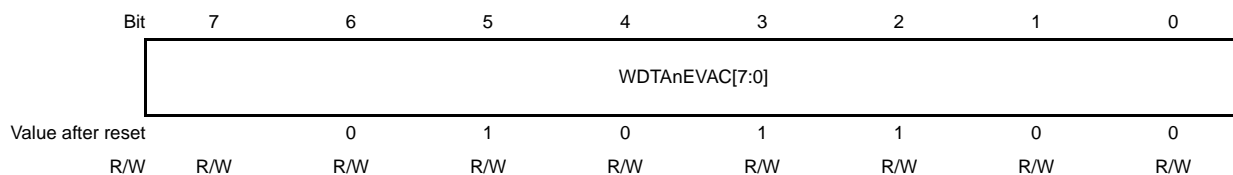


Table 21.12 WDTAnEVAC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|---|
| 7 to 0 | WDTAnEVAC [7:0] | Writing a variable activation code generates a WDTA trigger to control WDTAn count start and restart. Writing an incorrect value generates an error. After WDTAn starts, it cannot be stopped. When these bits are read or written, see Table 21.14, WDTAnEVAC Behavior |

The WDTAnEVAC7 bit is only valid if the VAC function is enabled (OPWDVAC = 1). **Table 21.13** lists the values of the WDTAnEVAC7 bit after reset according to the start-up options.

Table 21.13 WDTAnEVAC7 Value after a Reset

| Start-up options | | Start Mode | Value of WDTAnEVAC7 after a Reset |
|------------------|---------|------------------------|-----------------------------------|
| OPWDVAC | OPWDRUN | | |
| 1 | 1 | Default start | 1 |
| | 0 | Software trigger start | 0 |

The behavior of WDTAnEVAC during read/write accesses depends on the OPWDVAC setting, as shown in **Table 21.14**.

Table 21.14 WDTAnEVAC Behavior

| OPWDVAC | Description | WDTAnEVAC | |
|---------|--|--|---|
| | | Read | Write |
| 0 | The VAC function is disabled. WDTAnEVAC is disabled. | 2C _H is read. | Writing is ignored. |
| 1 | The VAC function is enabled. WDTAnEVAC is enabled. | 2C _H is read (before WDTAn starts in software trigger start mode). The variable activation code written last is returned (after WDTAn has started). | Write the variable activation code* ¹ . For details, see Section 21.5.2.1, Calculating an Activation Code when the VAC Function is Used . |

Note 1. Any other write value will cause an error.

21.3.4 WDTAnREF — WDTA Reference Value Register

This register contains the reference value for calculating the activation code of the VAC function. It is automatically updated after every trigger operation. For details, see **Section 21.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

If VAC function is disabled (OPWDVAC = 0), reading this register returns 00_H.

Access: This register can be read in 8-bit units.

Address: <WDTAn_base> + 0008_H

Value after reset: 00_H This register is initialized by a reset of any type.

| | | | | | | | | |
|-------------------|---------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | WDTAnREF[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 21.15 WDTAnREF Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 0 | WDTAnREF[7:0] | Reference value to calculate activation code of VAC function. |

21.3.5 WDTAnMD — WDTA Mode Register

This register specifies the overflow interval time, 75% interrupt enable/disable, and the window-open period.

The value of this register can be updated only once after release from the reset state and before the first WDTA trigger is generated. The updated value will be effective after the WDTA trigger register is written to.

When this register is updated after the first WDTA trigger has been generated, an error occurs. No error occurs when the same value is written to it.

Access: This register can be read/written in 8-bit units.

Address: <WDTAn_base> + 000C_H

Value after reset: The initial value depends on the start-up options (OPWDOVF[2:0]). Refer to **Table 21.7, WDTA Start-up Options**. This register is initialized by a reset of any type.

| | | | | | | | | |
|-------------------|---|---------------|-----|-----|----------|-----|--------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | WDTAnOVF[2:0] | | | WDTAnWIE | — | WDTAnWS[1:0] | |
| Value after reset | 0 | *1 | *1 | *1 | 0 | 1 | 1 | 1 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. The WDTAnOVF[2:0] value after reset can be set by the start-up options OPWDOVF[2:0].

Table 21.16 WDTAnMD Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---------------|--|----------------------------|------------|------------|------------------------|---|---|---|---------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 to 4 | WDTAnOVF[2:0] | Select the overflow interval time. <table border="1" data-bbox="675 1211 1426 1570"> <thead> <tr> <th>WDTAn OVF2</th> <th>WDTAn OVF1</th> <th>WDTAn OVF0</th> <th>Overflow interval time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>2⁹/ WDTATCKI</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2¹⁰/ WDTATCKI</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>2¹¹/ WDTATCKI</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2¹²/ WDTATCKI</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>2¹³/ WDTATCKI</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>2¹⁴/ WDTATCKI</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>2¹⁵/ WDTATCKI</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2¹⁶/ WDTATCKI</td> </tr> </tbody> </table> <ul style="list-style-type: none"> • WDTATCKI frequency can be selected from the following two count clocks by the start-up option OPBT0[21]. <ul style="list-style-type: none"> – High-speed mode (high-speed internal oscillation: 8 MHz) – Low-speed mode (high-speed internal oscillation/32: 250 kHz) | WDTAn OVF2 | WDTAn OVF1 | WDTAn OVF0 | Overflow interval time | 0 | 0 | 0 | 2 ⁹ / WDTATCKI | 0 | 0 | 1 | 2 ¹⁰ / WDTATCKI | 0 | 1 | 0 | 2 ¹¹ / WDTATCKI | 0 | 1 | 1 | 2 ¹² / WDTATCKI | 1 | 0 | 0 | 2 ¹³ / WDTATCKI | 1 | 0 | 1 | 2 ¹⁴ / WDTATCKI | 1 | 1 | 0 | 2 ¹⁵ / WDTATCKI | 1 | 1 | 1 | 2 ¹⁶ / WDTATCKI |
| WDTAn OVF2 | WDTAn OVF1 | WDTAn OVF0 | Overflow interval time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 2 ⁹ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 2 ¹⁰ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 2 ¹¹ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 2 ¹² / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 2 ¹³ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 2 ¹⁴ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 2 ¹⁵ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 2 ¹⁶ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | WDTAnWIE | Enables or disables the 75% interrupt request INTWDTAn: 0: INTWDTAn is disabled. 1: INTWDTAn is enabled. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 21.16 WDTAnMD Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|--------------|---|--------------|--------------|--------------------|---|---|-----|---|---|-----|---|---|-----|---|---|------|
| 1, 0 | WDTAnWS[1:0] | Select the window-open period: | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>WDTAn WS1</th> <th>WDTAn WS0</th> <th>Window-open Period</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>25%</td> </tr> <tr> <td>0</td> <td>1</td> <td>50%</td> </tr> <tr> <td>1</td> <td>0</td> <td>75%</td> </tr> <tr> <td>1</td> <td>1</td> <td>100%</td> </tr> </tbody> </table> | WDTAn WS1 | WDTAn WS0 | Window-open Period | 0 | 0 | 25% | 0 | 1 | 50% | 1 | 0 | 75% | 1 | 1 | 100% |
| WDTAn WS1 | WDTAn WS0 | Window-open Period | | | | | | | | | | | | | | | |
| 0 | 0 | 25% | | | | | | | | | | | | | | | |
| 0 | 1 | 50% | | | | | | | | | | | | | | | |
| 1 | 0 | 75% | | | | | | | | | | | | | | | |
| 1 | 1 | 100% | | | | | | | | | | | | | | | |

21.4 Interrupt Sources

The WDTA checks the status of the WDTA counter value, detects accesses to the WDTA-related registers, and generates an interrupt request. The following are WDTA interrupt requests:

- (1) INTWDTAn (WDTA timer count 75% interrupt request)
An interrupt request signal is generated at 75% of the counter overflow time of the WDTA timer.
An interrupt request signal can be enabled or disabled by using the WDTA mode register (WDTAnMD).
- (2) WDTAnTERR (WDTA error signal)
Detection of a WDTA error generates an error signal WDTAnTERR.
The WDTAnTERR signal is connected to the ECM module.

21.5 Function

21.5.1 WDTA after Release from the Reset State

21.5.1.1 Start Modes

There are two start modes (software start mode and default start mode) when WDTA starts after release from the reset state. The start mode can be selected by the start-up option. The start modes are listed in **Table 21.17**.

Table 21.17 Start Mode Selection

| Start-up Option | Start mode | Description |
|-----------------|--------------------|---|
| OPWDRUN | | |
| 0 | Software trigger | <ul style="list-style-type: none"> The WDTA counter stops (0000_H) after release from the reset state. Writing an activation code to the WDTA trigger register starts WDTA. |
| 1 | Default start mode | The WDTA counter starts after release from the reset state. |

21.5.1.2 Count Clock Selection

The count clock WDTATCKI can be selected by the start-up options.

The selection of the count clock WDTATCKI is listed in **Table 21.18**.

Table 21.18 Count Clock Selection

| Start-up Options | Mode | Count Clock |
|------------------|-----------------|--|
| OPWDMDS | | |
| 0 | High-speed mode | High-speed internal oscillation (8 MHz) |
| 1 | Low-speed mode | High-speed internal oscillation/32 (250 kHz) |

21.5.1.3 WDTA Settings after Release From The Reset State,

Table 21.19 shows the WDTA settings after release from the Reset State,.

Table 21.19 WDTA Settings after Release From The Reset State,

| Function | Setting | Remark |
|-----------------------------|-----------------------------------|--|
| Start mode | Specified by start-up options | |
| VAC function | | |
| Count clock selection | | |
| WDTA overflow interval time | Specified by start-up options | Modification is possible only once by the setting of the WDTA mode register (WDTAnMD). |
| 75% interrupt mode | 75% interrupt disabled | |
| Behavior on error detection | Error signal WDTAnTERR generation | |
| Window-open period | 100% | |

The setting of the WDTA mode register (WDTAnMD) is enabled when the first WDTA trigger is generated (writing an activation code to WDTAnWDTE and WDTAnEVAC). Change the setting of the WDTAnMD register before a WDTA trigger is generated.

Setting of WDTA by using WDTAnMD is possible only once. If the value set in WDTAnMD is changed after a WDTA trigger is generated, an error occurs. However, an error does not occur if the same value is set.

21.5.1.4 Default Start Mode Timing

The default start mode timing and the changes to the WDTA settings are illustrated in **Figure 21.2**.

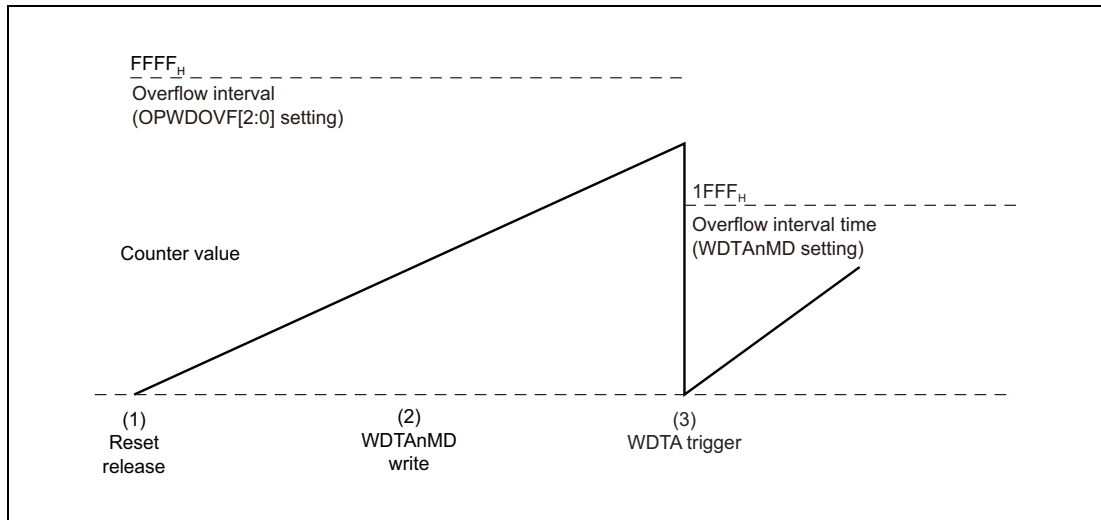


Figure 21.2 Timing Diagram of WDTA Start in Default Start Mode

The timing diagram shown in **Figure 21.2** shows the following behaviors:

- (1) In default start mode, the WDTA counter starts after release from the reset state. The overflow interval time after release from the reset state, is set by start-up options.
Example: Overflow interval time after release from the reset state,

$$= 2^{16}/\text{WDTATCKI} (\text{OPWDOVF}[2:0] = 111_{\text{B}})$$
- (2) WDTAnMD is set before a WDTA trigger is generated. Note that the setting is not applied immediately.
- (3) Write to the WDTA trigger register before the WDTA counter overflows. The WDTAnMD setting is applied by the generation of a WDTA trigger.
Example: Overflow interval time after a WDTA trigger is generated

$$= 2^{13}/\text{WDTATCKI}$$

21.5.1.5 Software Trigger Start Mode Timing

The software trigger start mode timing and the changes to the WDTA settings are illustrated in **Figure 21.3**.

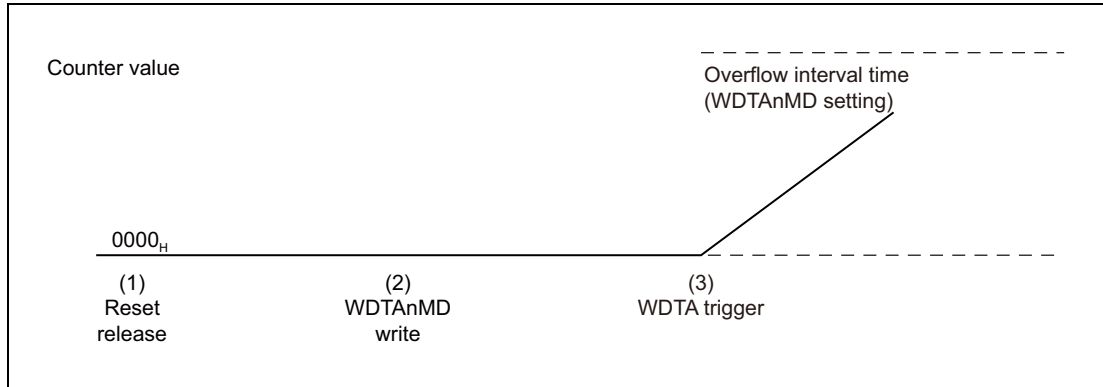


Figure 21.3 Timing Diagram of WDTA Start in Software Trigger Start Mode

The timing diagram above shows the following:

- (1) After release from the reset state, the counter remains 0000_H until the first trigger. The count clock is specified by the start-up options, but this setting has no effect because the counter is not operating.
- (2) WDTAnMD is written before the first trigger. However, the settings are not applied immediately.
- (3) The WDTA counter starts by the generation of a WDTA trigger. The overflow interval time and other settings specified in WDTAnMD are applied.

21.5.2 WDTA Trigger

Writing a special value called an activation code to the WDTA enable register (WDTAnWDTE) and the WDTA enable VAC register (WDTAnEVAC) leads to generation of a WDTA trigger.

The WDTA trigger has the following functions:

- Starting the WDTA counter in software trigger start mode
- Restarting the WDTA counter
- Setting the WDTA mode specified by the WDTAnMD register (only for the first WDTA trigger after release from the reset state,)

The WDTA trigger register, which generates a WDTA trigger, is specified by the start-up option OPWDVAC.

Table 21.20 lists the WDTA trigger registers and activation codes.

Table 21.20 WDTA Trigger Registers and Activation Codes

| Type of activation code | Trigger register | Activation code |
|-------------------------|------------------|--|
| Fixed (OPWDVAC = 0) | WDTAnWDTE | AC _H |
| Variable (OPWDVAC = 1) | WDTAnEVAC | For details, refer to Section 21.5.2.1, Calculating an Activation Code when the VAC Function is Used. |

21.5.2.1 Calculating an Activation Code when the VAC Function is Used

Use the following expression to calculate the variable activation code (ExpectWDTE) to be set to the WDTA trigger register (WDTAnEVAC) when the VAC function is used, by using the WDTA reference value register (WDTAnREF):

$$\text{ExpectWDTE} = \text{AC}_H - \text{WDTAnREF (old)}$$

Note that the value in the WDTAnREF register is updated every time a activation code is written to the trigger register WDTAnEVAC. Use the following expression to calculate the updated value of the WDTAnREF register:

$$\text{WDTAnREF (new)} = (\text{rotate the value of ExpectWDTE to the left by 1 bit})$$

Table 21.21 lists the variable activation codes according to the number of WDTA triggers.

Table 21.21 Expected Variable Activation Code Development

| No ^{*1} | WDTAnREF (old) | | ExpectWDTE (AC _H - WDTAnREF) | | WDTAnREF (new) | |
|------------------|----------------|-----------------|--|-----------------|----------------|-----------------|
| 0 | 0000 0000 | 00 _H | 1010 1100 | AC _H | 0101 1001 | 59 _H |
| 1 | 0101 1001 | 59 _H | 0101 0011 | 53 _H | 1010 0110 | A6 _H |
| 2 | 1010 0110 | A6 _H | 0000 0110 | 06 _H | 0000 1100 | 0C _H |
| ... | ... | ... | ... | ... | ... | ... |

Note 1. Number of triggers after reset

NOTE

Writing an incorrect activation code generates an error.

21.5.3 WDTA Error Detection

WDTA detects an error, including generation of WDTA count overflow or illegal operations.

The following events are detected as errors:

- WDTA counter overflow
- Wrong activation code is written to the WDTA trigger register
- Writing to the trigger register at a time outside the window-open period
- When it is attempted to change the setting value of the WDTA mode register WDTAnMD after the first WDTA trigger has been generated
- When updating the setting value of the WDTA mode register WDTAnMD twice before the first WDTA trigger is generated

Figure 21.4 shows generation of an error signal WDTAnTERR when default mode is selected and the counter overflows.

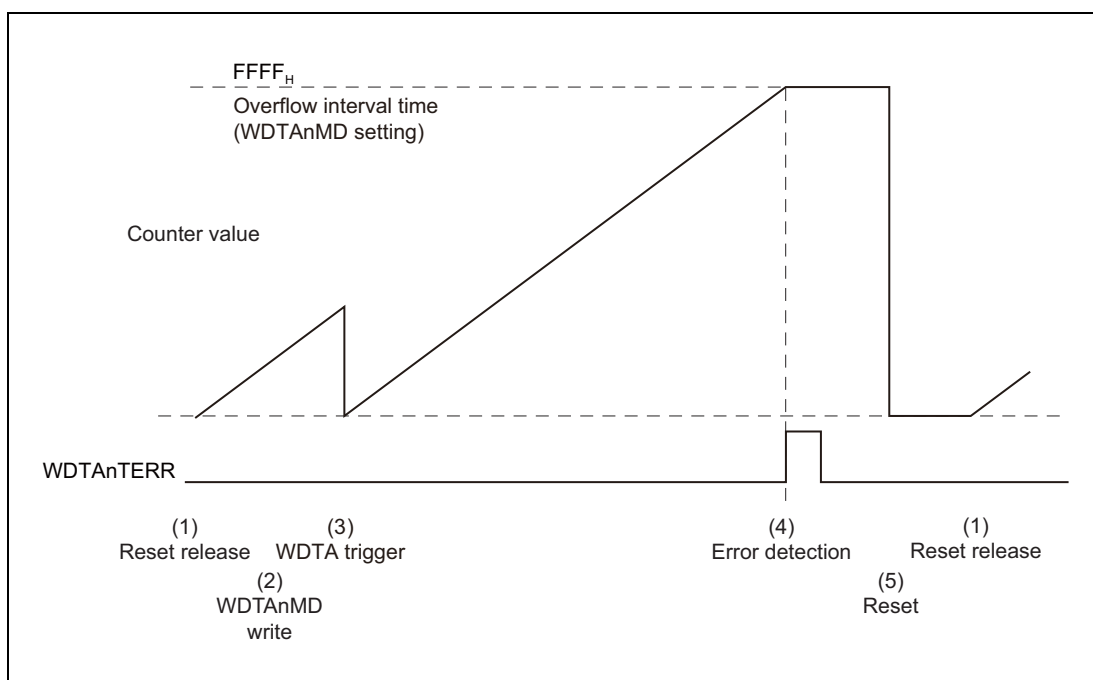


Figure 21.4 Timing Diagram of WDTA Error Signal WDTAnTERR Generation

The timing diagram above shows the following:

- (1) In default start mode, the WDTA counter starts after release from the reset state. The overflow interval time after release from the reset state, is set by start-up options.
- (2) WDTAnMD is set before a WDTA trigger is generated. In this case, $2^{16}/\text{WDTATCKI}$ is set for the overflow interval time.
- (3) The WDTAnMD setting is applied due to the WDTA trigger.
- (4) When the counter overflows, an error is detected. An error signal WDTAnTERR is generated. The counter value remains until a reset is performed.
- (5) When the reset is performed, the counter is cleared and stopped until release from the reset state.

21.5.4 75% Interrupt Request Signals

When the WDTA counter reaches 75% of the time set for the overflow interval, the interrupt request INTWDTAn is generated.

By use of WDTAnMD.WDTAnWIE this function can be enabled or disabled afterwards.

Figure 21.5 shows the 75% interrupt request generation under following conditions:

- Default start mode selected
- 75% interrupt request is enabled after the first WDTA trigger is generated
- WDTA overflow interval time: $2^{16}/\text{WDTATCKI}$

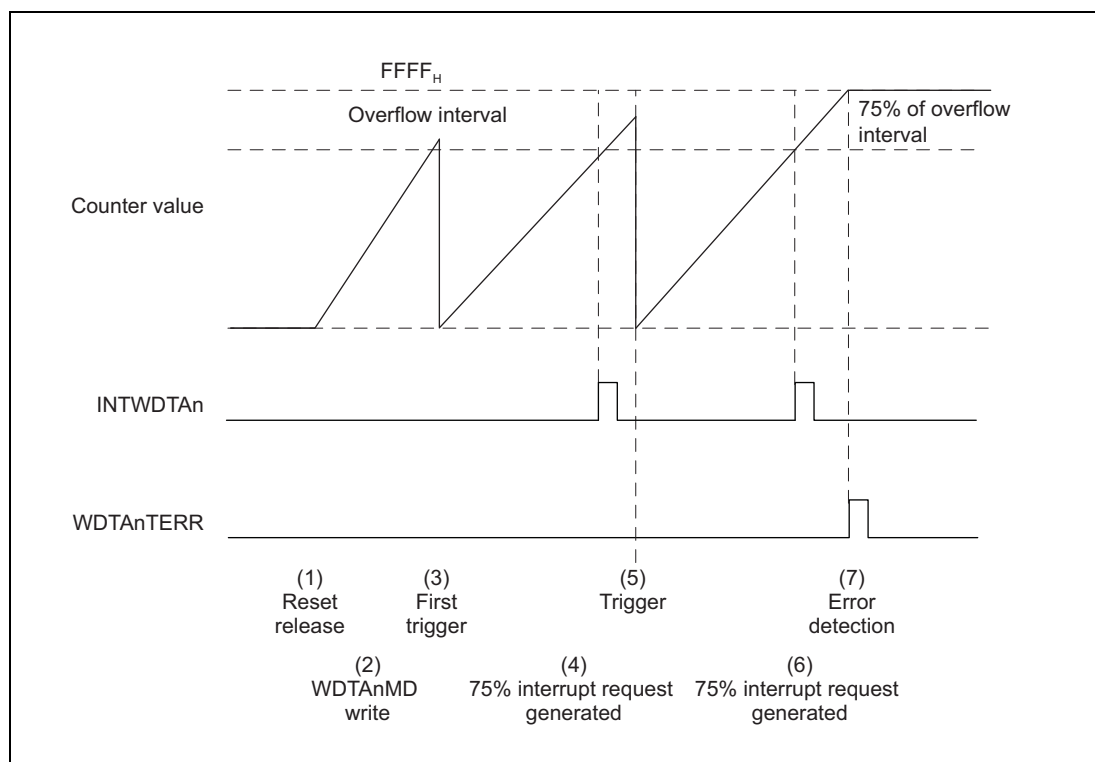


Figure 21.5 Timing Diagram of WDTA 75% Interrupt Request Signals

- (1) In default start mod, WDTA count starts after reset is released.
The overflow interval time period after reset is released is set by the start-up options.
- (2) WDTAnMD is set before the WDTA trigger is generated. In this figure, $2^{16}/\text{WDTATCKI}$ is set as the overflow interval time period.
- (3) The WDTAnMD setting is applied at the WDTA trigger.
- (4) When the WDTA counter reaches 75% of the set overflow interval time, an interrupt request INTWDTAn is generated.
- (5) The count restarts at the WDTA trigger.
- (6) When the WDTA counter reaches 75% of the set overflow interval time, an interrupt request INTWDTAn is generated.
- (7) When the counter overflows, an error is detected. An error signal WDTAnTERR is generated.
The counter value remains unchanged until a reset is performed.

21.5.5 Window Function

The period when a WDTA trigger is valid (window-open period) can be set. If the window-open period is set to the value less than 100%, an error occurs by the WDTA trigger generated not in the window-open period. The window-open period after release from the reset state, is 100%. The period is set to the value by the WDTAnMD.WDTAnWS[1:0] setting after the first WDTA trigger is generated.

Figure 21.6 shows window function operations under the following conditions.

- Default start mode is selected.
- The 25% window open period is valid (WDTAnWS[1:0] = 00_B) after the first WDTA trigger
- WDTA overflow interval time period: $2^{16}/\text{WDTATCKI}$

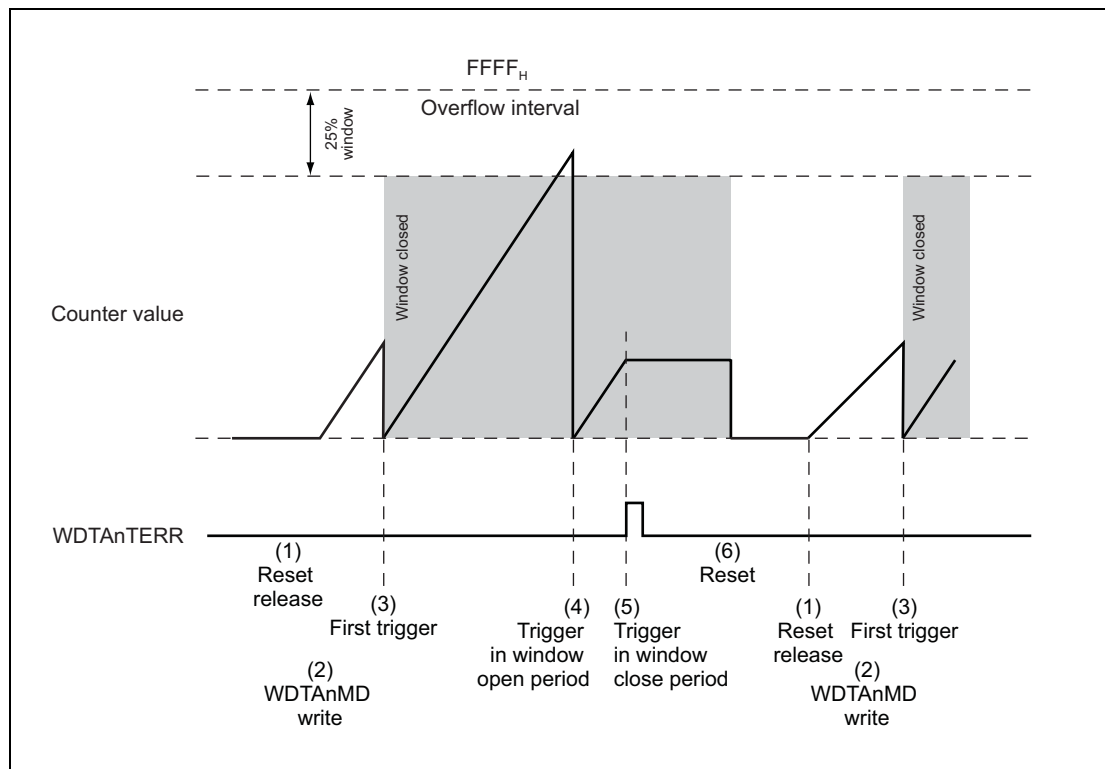


Figure 21.6 Timing Diagram of WDTA Window Function

- (1) In default start mode, WDTA count starts after reset is released.
The overflow interval time period after reset is released is set by the start-up options.
- (2) WDTAnMD is set before the WDTA trigger is generated. In this figure, $2^{16}/\text{WDTATCKI}$ is set as the overflow interval time period.
- (3) The WDTAnMD setting is applied at the WDTA trigger.
- (4) WDTA count restarts at the WDTA trigger during the window open period.
- (5) An error is detected at the WDTA trigger during the window close period.
An error signal WDTAnTERR is generated. The counter value remains unchanged until a reset is performed.
- (6) When the reset is performed, the counter is cleared and stops until the reset is released.

Section 22 OS Timer (OSTM)

The OS Timer (OSTM) is a 32-bit timer/counter. OSTM is implemented 7 units.

OSTM can be used in interval timer mode or in free-running comparison mode. The settings for operating mode specify the direction of counting (up or down) in control of the generation of interrupt requests.

The OSTM is synchronized with other peripheral functions by the count clock enable signal and the count start signal.

22.1 Features of RH850/P1M-E OSTM

22.1.1 Number of Units

This microcontroller has the following number of units of the OSTM_n.

Table 22.1 Number of Units

| Product | RH850/P1M-E |
|-----------------|----------------------------------|
| Number of Units | 7 |
| Name | OSTM0, OSTM1, and OSTM3 to OSTM7 |

Table 22.2 Index

| Index | Meaning |
|-------|--|
| n | Throughout this section, the individual OSTM _n units are identified by the index "n": for example, OSTM _n CNT is the OSTM _n counter register. |

22.1.2 Register Base Address

OSTMn base addresses are listed in the following table.

OSTMn register addresses are given as offsets from these base addresses.

Table 22.3 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <OSTM0_base> | FFDD 8000 _H |
| <OSTM1_base> | FFDD 9000 _H |
| <OSTM3_base> | FFD7 0000 _H |
| <OSTM4_base> | FFD7 0040 _H |
| <OSTM5_base> | FFD7 0080 _H |
| <OSTM6_base> | FFD7 00C0 _H |
| <OSTM7_base> | FFD7 0100 _H |

22.1.3 Clock Supply

Clock supply by and to OSTMn is listed in the following table.

Table 22.4 Clock Supply

| Unit Name | Clock for the Unit | Supply Clock Name |
|-----------|--------------------|---------------------------------------|
| OSTM0 | PCLK | High speed peripheral clock CLK_HSB*1 |
| OSTM1 | PCLK | High speed peripheral clock CLK_HSB*1 |
| OSTM3 | PCLK | High speed peripheral clock CLK_HSB |
| OSTM4 | PCLK | High speed peripheral clock CLK_HSB |
| OSTM5 | PCLK | High speed peripheral clock CLK_HSB |
| OSTM6 | PCLK | High speed peripheral clock CLK_HSB |
| OSTM7 | PCLK | High speed peripheral clock CLK_HSB |

Note 1. OSTMnTCKE is selectable as the counter source clock.

22.1.4 Interrupt Requests

OSTMn interrupt requests are listed in the following table.

Table 22.5 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|-----------------|------------------|------------------------|
| OSTM0 | | | |
| INTOSTM0 | OSTM0 interrupt | 74 | — |
| OSTM1 | | | |
| INTOSTM1 | OSTM1 interrupt | 75 | — |
| OSTM3 | | | |
| INTOSTM3 | OSTM3 interrupt | (FEINT) | — |
| OSTM4 | | | |
| INTOSTM4 | OSTM4 interrupt | (FEINT) | — |
| OSTM5 | | | |
| INTOSTM5 | OSTM5 interrupt | (FEINT) | — |
| OSTM6 | | | |
| INTOSTM6 | OSTM6 interrupt | (FEINT) | — |
| OSTM7 | | | |
| INTOSTM7 | OSTM7 interrupt | (FEINT) | — |

22.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

22.1.6 External Input/Output Signals

An OSTMn input/output signals are listed in the following table.

Table 22.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|------------------|-----|--------------|----------------------------------|
| OSTM0: | | | |
| OSTM0TTOUT | O | OSTM0 output | OSTM0O |
| OSTM1: | | | |
| OSTM1TTOUT | O | OSTM1 output | OSTM1O |

22.2 Overview

Each OSTMn is a 32-bit timer/counter.

OSTMn can be used in interval timer mode or in free-running comparison mode. The settings for operating mode determine the direction of counting (up or down) that is used for the generation of interrupt requests.

The OSTMn is synchronized with other peripheral modules by the counter-clock-enable signal (OSTMnTCKE) and the count start signal (OSTMnTSST).

22.2.1 Functional Overview

The OSTMn has the following features.

- Two operating modes
 - Interval timer mode
 - Free-running comparison mode
- INTOSTMn interrupt
- Generation of an output signal from the OSTMnO pin (OSTM0, OSTM1)
 - Software control mode
 - Timer output toggle mode
- Using the timing protection timer (TPT) to monitor times
 - Monitoring of the times taken in processing tasks and interrupts is possible by using INTOSTM3 to INTOSTM7 assigned as FEINT interrupts.

22.2.2 Block Diagram

The following block diagram shows the main components of the OSTMn.

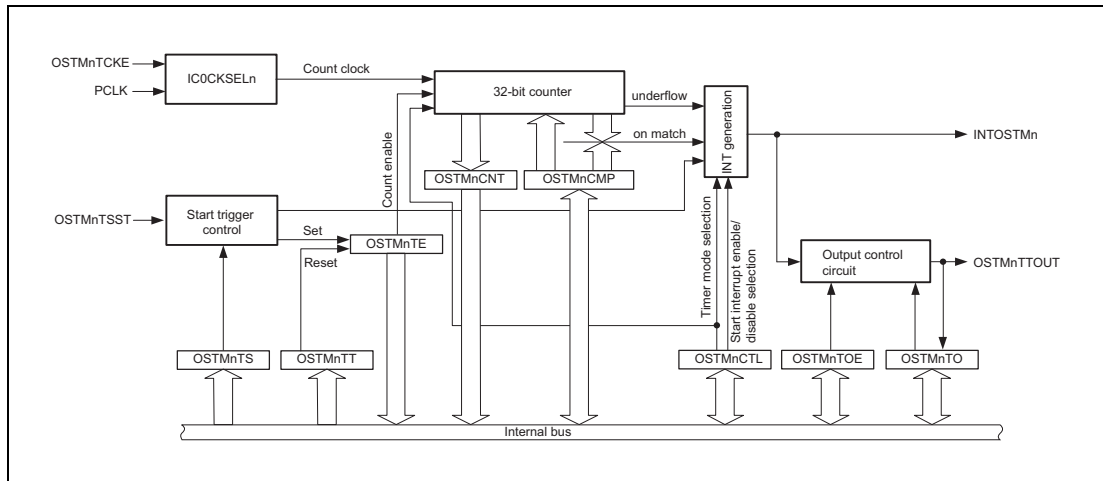


Figure 22.1 Block Diagram of the OSTMn (OSTM0, OSTM1)

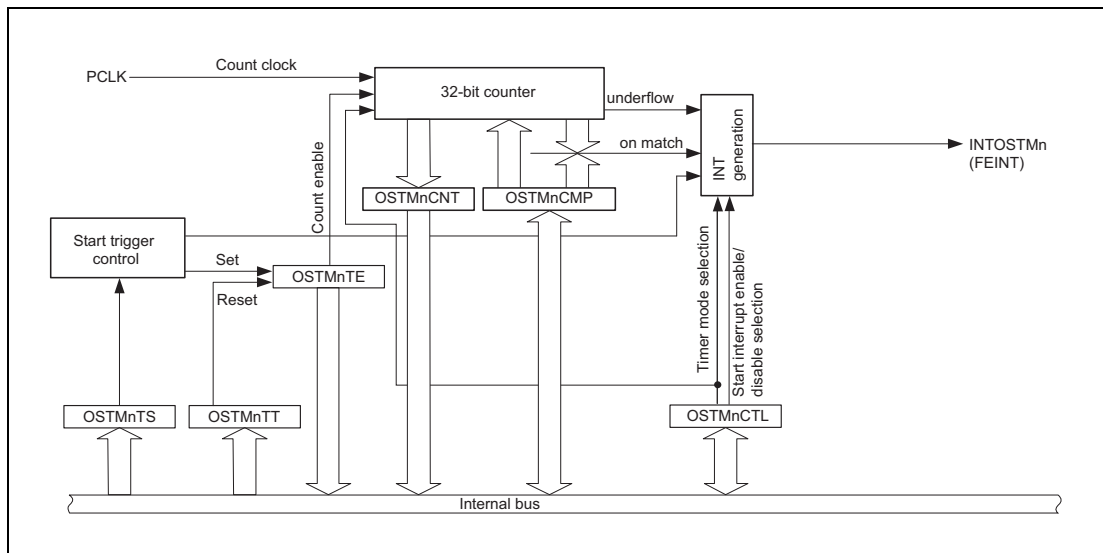


Figure 22.2 Block Diagram of the OSTMn (OSTM3 to OSTM7)

22.2.3 Counter clock

The clock signals for counting by OSTM0 and OSTM1 are defined by PCLK and OSTMnTCKE in the following ways. The count signals for counting by OSTM3 to OSTM7 are always defined by PCLK.

- If C0KSELn.IC0TMENn = 0, the OSTMnTCKE signal being at the high level (logical 1) selects counting of cycles of PCLK.
- IC0KSELn.IC0TMENn being set to 1 selects the TAUDn or TAUJn signal as the counter-clock-enable signal, OSTMnTCKE. The cycles of the clock signals for counting by OSTMn are synchronized with the cycles of the clock signals generated as TAUDn or TAUJn (any from among CK0 to CK3).

This is illustrated in the following figures.

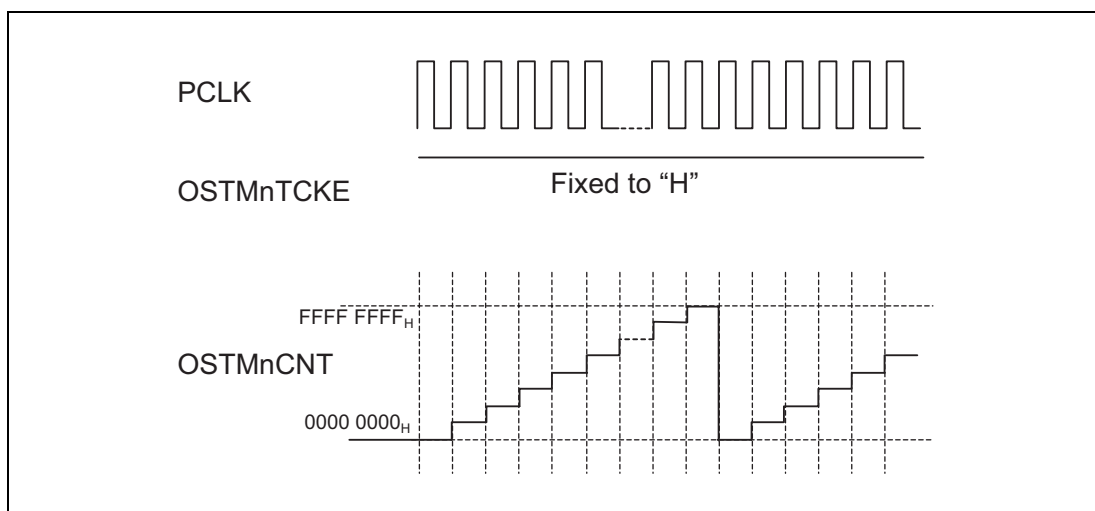


Figure 22.3 Counting when OSTMnTCKE is Fixed to H

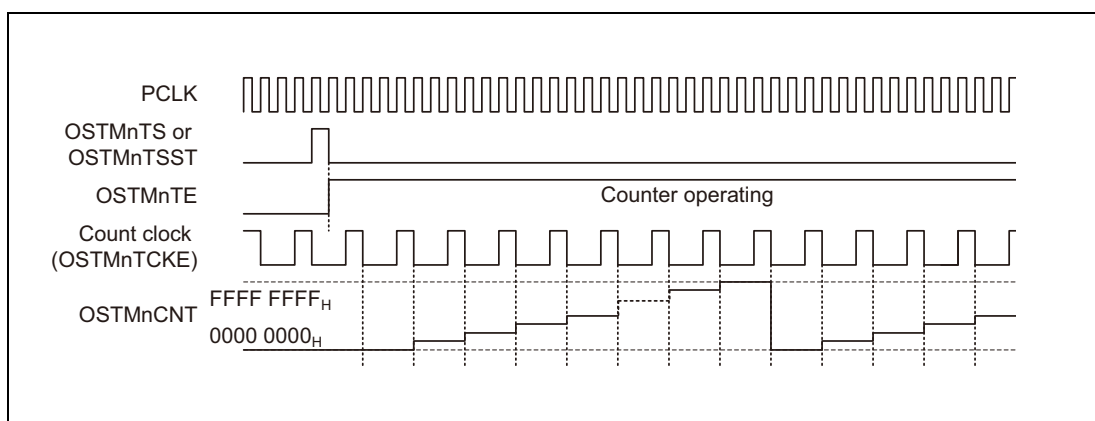


Figure 22.4 Counting in Response to Transitions of the OSTMnTCKE Input Signal

The counter-clock-enable signal (OSTMnTCKE) for each OSTMn can be selected by the IC0CKSEL0 or IC0CKSEL1 register from among 40 channels: 16 channels from each of TAUD0 and TAUD1, and 4 channels from each of TAUJ0 and TAUJ1.

For the counter-clock-enable signals for TAUDn and TAUJn (TAUDnTCKENm and TAUJnTCKENm) and the clock signals generated in TAUDn and TAUJn (CK0 to CK3), see **Figure 22.2** and **Figure 23.2**.

CAUTIONS

1. Select a counter-clock-enable signal for OSTMn while the operation of OSTMn is stopped (the OSTMnTE.OSTMnTE bit is 0).
2. After using the IC0CKSELn.IC0TMSELn[1:0], IC0CKSELn.IC0CKSELn3[1:0], IC0CKSELn.IC0CKSELn2[1:0], IC0CKSELn.IC0CKSELn1[3:0], and IC0CKSELn.IC0CKSELn0[3:0] bits to select the counter-clock-enable signal for OSTMn, set the IC0CKSELn.IC0TMENn bit to 1.
3. If TAUDn or TAUJn is selected as the source of the counter-clock-enable signal for OSTMn (the IC0CKSELn.IC0TMENn bit is 1), do not change the operation of TAUDn or TAUJn while the OSTMn is operating (the OSTMnTE.OSTMnTE bit is 1).

[Setting procedure]

- (1) Confirm that the OSTMnTE.OSTMnTE bit is 0 (OSTMn operation is stopped).
 - (2) Set the IC0CKSELn.IC0TMSELn[1:0], IC0CKSELn.IC0CKSELn3[1:0], IC0CKSELn.IC0CKSELn2[1:0], IC0CKSELn.IC0CKSELn1[3:0], and IC0CKSELn.IC0CKSELn0[3:0] bits to select the counter-clock enable signal for the OSTMn.
 - (3) Set the IC0CKSELn.IC0TMENn bit to 1.
 - (4) Enable OSTMn operation by setting the OSTMnTS.OSTMnTS bit to 1.
4. Operating system timers OSTM3 to OSTM7 do not have IC0CKSELn registers. Only PCLK is available as the clock for counting.
-

22.2.4 Output Modes (OSTM0 and OSTM1)

The OSTMn has the following output modes. The mode is selected by the setting of the OSTMnTOE.OSTMnTOE bit.

- Software control mode (the OSTMnTOE.OSTMnTOE bit is 0): The value set in the OSTMnTO.OSTMnTO bit is output to OSTMnTTOUT.
- Timer-output toggling mode (the OSTMnTOE.OSTMnTOE bit is 1): The OSTMnTTOUT output is toggled each time an INTOSTMn request is generated.

Both output modes are illustrated in the following figure.

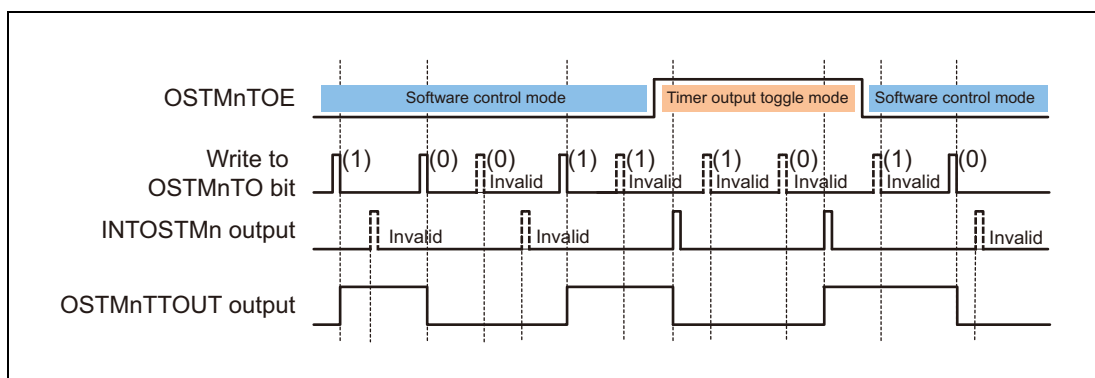


Figure 22.5 Timing Diagram of Output Modes

The above timing diagram shows the following operations.

- In software control mode, the level of the OSTMnTTOUT output changes in accord with the value set in the OSTMnTO.OSTMnTO bit.
- In timer-output toggling mode, the value of the OSTMnTO.OSTMnTO bit and level of the OSTMnTTOUT output are toggled each time an INTOSTMn interrupt request is generated.

22.2.5 Interrupt Requests (INTOSTMn)

An INTOSTMn interrupt request is generated on counter underflow (interval timer mode) or when the counter matches the compare value (free-running comparison mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

The INTOSTMn is the trigger for toggling of the OSTMnTTOUT output in the timer output toggling mode (OSTMnTOE.OSTMnTOE = 1), so the setting of the OSTMnCTL.OSTMnMD0 bit also affects OSTMnTTOUT output.

This is illustrated in the following figure.

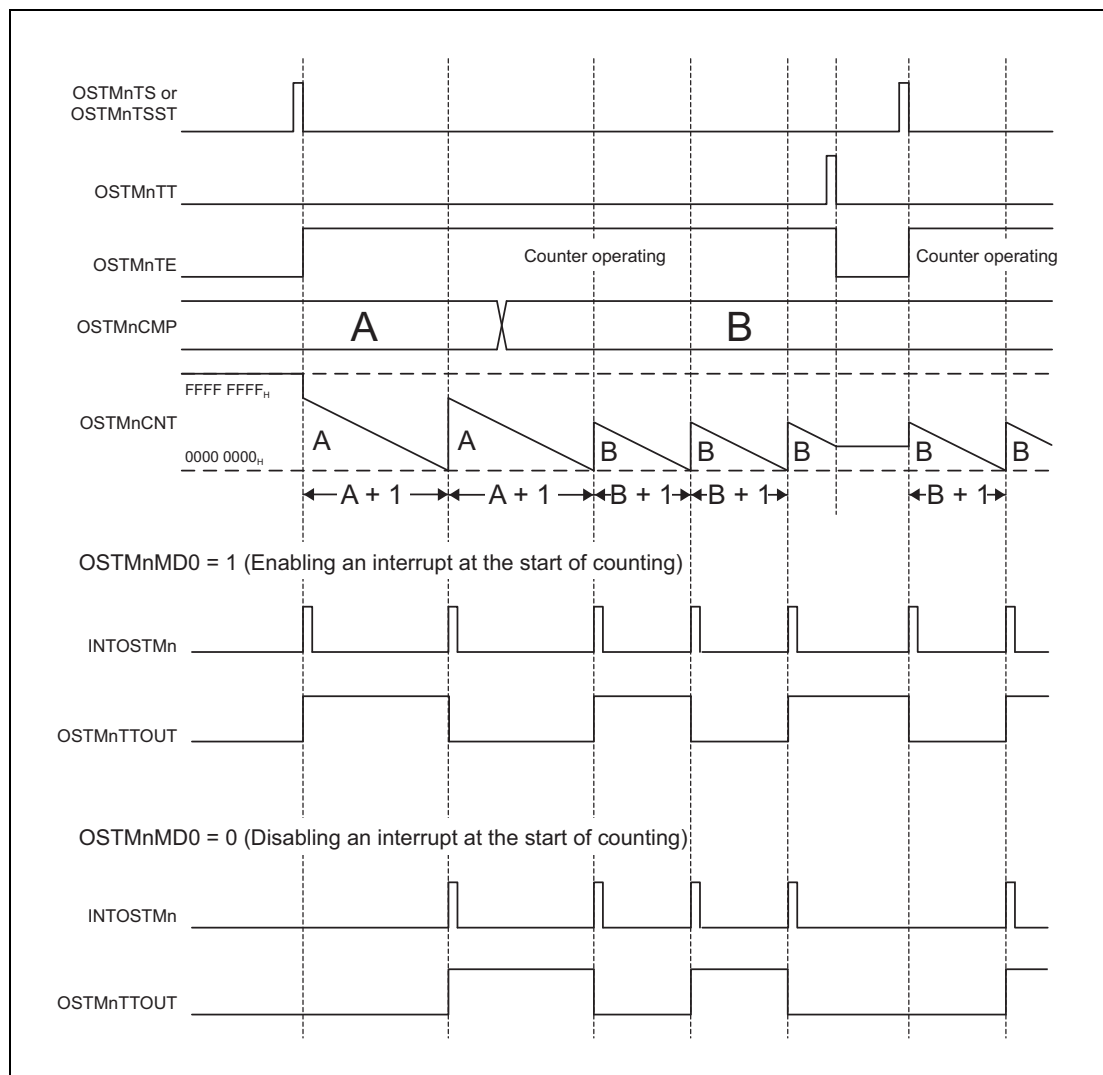


Figure 22.6 Generating an Interrupt when Counting Starts (Interval Timer Mode)

22.3 Registers

22.3.1 List of Registers

OSTMn registers are listed in the following table.

For details about <OSTMn_base>, see **Section 22.1.2, Register Base Address**.

Table 22.7 List of Registers

| Module Name | Register Name | Symbol | Address |
|-------------|------------------------------------|-----------|--------------------------------|
| OSTMn | OSTMn compare register | OSTMnCMP | <OSTMn_base> + 00 _H |
| OSTMn | OSTMn counter register | OSTMnCNT | <OSTMn_base> + 04 _H |
| OSTMn | OSTMn output register | OSTMnTO | <OSTMn_base> + 08 _H |
| OSTMn | OSTMn output enable register | OSTMnTOE | <OSTMn_base> + 0C _H |
| OSTMn | OSTMn count enable status register | OSTMnTE | <OSTMn_base> + 10 _H |
| OSTMn | OSTMn count start trigger register | OSTMnTS | <OSTMn_base> + 14 _H |
| OSTMn | OSTMn count stop trigger register | OSTMnTT | <OSTMn_base> + 18 _H |
| OSTMn | OSTMn control register | OSTMnCTL | <OSTMn_base> + 20 _H |
| OSTM | OSTM0 clock select register | IC0CKSEL0 | FFDD 6000 _H |
| OSTM | OSTM1 clock select register | IC0CKSEL1 | FFDD 6004 _H |

22.3.2 Details of OSTMn Registers

22.3.2.1 OSTMnCMP — OSTMn Compare Register

This register stores the start value of the down-counter or the value with which the counter is compared, depending on the mode of operation.

Access: This register can be read/written in 32-bit units.

Address: <OSTMn_base> + 00_H

Value after reset: 0000 0000_H. This register is initialized by a reset of any type.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | OSTMnCMP[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OSTMnCMP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 22.8 OSTMnCMP Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|---|
| 31 to 0 | OSTMnCMP [31:0] | <ul style="list-style-type: none"> In interval timer mode: start value of the down-counter In free-running comparison mode: compare value |

22.3.2.2 OSTMnCNT — OSTMn Counter Register

This register indicates the counter value of the timer.

Access: This register can be read in 32-bit units.

Address: <OSTMn_base> + 04_H

Value after reset: FFFF FFFF_H. This register is initialized by a reset of any type.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | OSTMnCNT[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OSTMnCNT[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 22.9 OSTMnCNT Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---------------------|
| 31 to 0 | OSTMnCNT [31:0] | Timer counter value |

Table 22.10 shows the correspondence between OSTM operating mode, counting direction and start value of the OSTMn. The start value is the value read from the counter after a change to the operating mode.

Table 22.10 Correspondence between Operating Mode, Counting Direction and Start Value

| Timer operating mode | OSTMnCTL.OSTMnMD1 | Counting direction | Start value |
|------------------------------|-------------------|--------------------|------------------------|
| Interval timer mode | 0*1 | Down | FFFF FFFF _H |
| Free-running comparison mode | 1 | Up | 0000 0000 _H |

Note 1. Value after reset.

OSTMnCNT is loaded with the value in the OSTMnCMP register if it is in interval timer mode or with the counter value 0000 0000_H if it is in free-running comparison mode at the start of a cycle of counting after OSTMnTE has become 1.

22.3.2.3 OSTMnTO — OSTMn Output Register

The OSTM Output Register is used to specify and read the level of an OSTMnTTOUT output signal. The setting of this register is only valid in OSTMn (n = 0, 1).

Access: This register can be read/written in 8-bit units. Writing can only proceed when the software control mode is selected (OSTMnTOE.OSTMnTOE = 0).

Address: <OSTMn_base>+ 08_H

Value after reset: 00_H
This register is initialized by a reset from any source.

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | OSTMnTO |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 22.11 OSTMnTO Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | OSTMnTO | This bit specifies or indicates the level of the OSTMnTTOUT output signal. 0: Low level 1: High level |

22.3.2.4 OSTMnTOE — OSTMn Output Enable Register

The OSTM Output Enable Register specifies OSTMnTTOUT output mode. The setting of this register is only valid in OSTMn (n = 0, 1).

Access: This register can be read/written in 8-bit units.

Address: <OSTMn_base>+ 0C_H

Value after reset: 00_H
This register is initialized by a reset from any source.

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | OSTMnTOE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 22.12 OSTMnTOE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | OSTMnTOE | This bit specifies the OSTMnTTOUT output mode. 0: Software control mode: The level corresponding to the setting of OSTMnTO.OSTMnTO is output to OSTMnTTOUT. 1: Timer-output toggling mode: OSTMnTTOUT output is toggled whenever an INTOSTMn interrupt request is generated. |

22.3.2.5 OSTMnTE — OSTMn Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access: This register can only be read in 8-bit units.

Address: <OSTMn_base> + 10_H

Value after reset: 00_H. This register is initialized by a reset of any type.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | OSTMnTE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 22.13 OSTMnTE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | OSTMnTE | Indicates whether the counter is enabled or disabled: 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1, or to OSTMnTSST being 1. Setting OSTMnTT.OSTMnTT to 1 resets this bit to 0. |

NOTE

If the counter is disabled, the counter OSTMnCNT retains its value.

If the counter is restarted:

- It restarts counting down from the value in the OSTMnCMP register if it is in interval timer mode or
- It restarts counting up from the counter value 0000 0000_H if it is in free-running comparison mode.

22.3.2.6 OSTMnTS — OSTMn Count Start Trigger Register

This register starts the counter.

Access: This register can only be written in 8-bit units. It is always read as 00_H.

Address: <OSTMn_base> + 14_H

Value after reset: 00_H. This register is initialized by a reset of any type.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | OSTMnTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 22.14 OSTMnTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | OSTMnTS | Starts the counter: 0: This setting is invalid. 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. <ul style="list-style-type: none"> • In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. • In free-running comparison mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1. |

22.3.2.7 OSTMnTT — OSTMn Count Stop Trigger Register

This register stops the counter.

Access: This register can only be written in 8-bit units. It is always read as 00_H.

Address: <OSTMn_base> + 18_H

Value after reset: 00_H. This register is initialized by a reset of any type.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | OSTMnTT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 22.15 OSTMnTT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | OSTMnTT | Stops the counter: 0: This setting is invalid. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit. |

22.3.2.8 OSTMnCTL — OSTMn Control Register

This register specifies the operating mode for the counter and controls the generation of INTOSTMn interrupt requests when counting starts.

Although this register is readable and writable, writing to it is only possible when OSTMnTE.OSTMnTE = 0; that is, the register becomes read-only when OSTMnTE.OSTMnTE = 1.

Access: This register can be read/written in 8-bit units.

Address: <OSTMn_base> + 20_H

Value after reset: 00_H. This register is initialized by a reset of any type.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|----------|----------|
| | — | — | — | — | — | — | OSTMnMD1 | OSTMnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 22.16 OSTMnCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | OSTMnMD1 | Specifies the operating mode for the counter: 0: Interval timer mode 1: Free-running comparison mode |
| 0 | OSTMnMD0 | Controls the generation of INTOSTMn interrupt requests at the start of counting: 0: Interrupts when counting starts are disabled. 1: Interrupts when counting starts are enabled. |

22.3.2.9 IC0CKSEL0 — OSTM0 Clock Select Register

This register selects the counter-clock-enable signal as the clock source to drive counting for the OSTM0 counter.

Access: This register can be read/written in 16-bit units.

Address: FFDD 6000_H

Value after reset: 0000_H. This register is initialized by a reset of any type.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|----|-----------------|------------------|------------------|------------------|-----|-----|------------------|-----|-----|-----|-----|-----|-----|-----|
| | IC0TMEN0 | — | IC0TMSEL0 [1:0] | IC0CKSEL03 [1:0] | IC0CKSEL02 [1:0] | IC0CKSEL01 [3:0] | | | IC0CKSEL00 [3:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 22.17 IC0CKSEL0 Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | |
|-----------------|------------------------------|---|-----------------|------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 15 | IC0TMEN0 | Selects the counter-clock-enable signal for OSTM0. 0: The counter-clock-enable signal is fixed to 1 (PCLK is selected as the source of the clock signal for counting) 1: Selects the peripheral module selected by IC0TMSEL0[1:0] as the counter-clock-enable signal. | | | | | | | | | | |
| 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | |
| 13, 12 | IC0TMSEL0 [1:0] | Select the counter-clock-enable signal for the OSTM0 counter (Enabled only when IC0TMEN0 = 1). | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0TMSEL0[1:0]</th> <th>Selected Peripheral Function</th> </tr> </thead> <tbody> <tr> <td>00_B</td> <td>TAUD0</td> </tr> <tr> <td>01_B</td> <td>TAUD1</td> </tr> <tr> <td>10_B</td> <td>TAUJ0</td> </tr> <tr> <td>11_B</td> <td>TAUJ1</td> </tr> </tbody> </table> | IC0TMSEL0[1:0] | Selected Peripheral Function | 00 _B | TAUD0 | 01 _B | TAUD1 | 10 _B | TAUJ0 | 11 _B | TAUJ1 |
| IC0TMSEL0[1:0] | Selected Peripheral Function | | | | | | | | | | | |
| 00 _B | TAUD0 | | | | | | | | | | | |
| 01 _B | TAUD1 | | | | | | | | | | | |
| 10 _B | TAUJ0 | | | | | | | | | | | |
| 11 _B | TAUJ1 | | | | | | | | | | | |
| 11, 10 | IC0CKSEL03 [1:0] | Select the counter-clock-enable signal for the OSTM0 counter (Enabled only when IC0TMEN0 = 1 and IC0TMSEL0[1:0] = 11 _B). | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0CKSEL03[1:0]</th> <th>Selected TAUJ1 Channel</th> </tr> </thead> <tbody> <tr> <td>00_B</td> <td>TAUJ1 channel 0</td> </tr> <tr> <td>01_B</td> <td>TAUJ1 channel 1</td> </tr> <tr> <td>10_B</td> <td>TAUJ1 channel 2</td> </tr> <tr> <td>11_B</td> <td>TAUJ1 channel 3</td> </tr> </tbody> </table> | IC0CKSEL03[1:0] | Selected TAUJ1 Channel | 00 _B | TAUJ1 channel 0 | 01 _B | TAUJ1 channel 1 | 10 _B | TAUJ1 channel 2 | 11 _B | TAUJ1 channel 3 |
| IC0CKSEL03[1:0] | Selected TAUJ1 Channel | | | | | | | | | | | |
| 00 _B | TAUJ1 channel 0 | | | | | | | | | | | |
| 01 _B | TAUJ1 channel 1 | | | | | | | | | | | |
| 10 _B | TAUJ1 channel 2 | | | | | | | | | | | |
| 11 _B | TAUJ1 channel 3 | | | | | | | | | | | |
| 9, 8 | IC0CKSEL02 [1:0] | Select the counter-clock-enable signal for the OSTM0 counter (Enabled only when IC0TMEN0 = 1 and IC0TMSEL0[1:0] = 10 _B). | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0CKSEL02[1:0]</th> <th>Selected TAUJ0 Channel</th> </tr> </thead> <tbody> <tr> <td>00_B</td> <td>TAUJ0 channel 0</td> </tr> <tr> <td>01_B</td> <td>TAUJ0 channel 1</td> </tr> <tr> <td>10_B</td> <td>TAUJ0 channel 2</td> </tr> <tr> <td>11_B</td> <td>TAUJ0 channel 3</td> </tr> </tbody> </table> | IC0CKSEL02[1:0] | Selected TAUJ0 Channel | 00 _B | TAUJ0 channel 0 | 01 _B | TAUJ0 channel 1 | 10 _B | TAUJ0 channel 2 | 11 _B | TAUJ0 channel 3 |
| IC0CKSEL02[1:0] | Selected TAUJ0 Channel | | | | | | | | | | | |
| 00 _B | TAUJ0 channel 0 | | | | | | | | | | | |
| 01 _B | TAUJ0 channel 1 | | | | | | | | | | | |
| 10 _B | TAUJ0 channel 2 | | | | | | | | | | | |
| 11 _B | TAUJ0 channel 3 | | | | | | | | | | | |

Table 22.17 IC0CKSEL0 Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | |
|-------------------|------------------------|--|-----------------|------------------------|-------------------|-----------------|-------------------|-----------------|-------------------|-----------------|---|--|-------------------|------------------|-------------------|------------------|-------------------|------------------|
| 7 to 4 | IC0CKSEL01 [3:0] | Select the counter-clock-enable signal for the OSTM0 counter (Enabled only when IC0TMEN0 = 1 and IC0TMSEL0[1:0] = 01 _B). | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0CKSEL01[3:0]</th> <th>Selected TAUD1 Channel</th> </tr> </thead> <tbody> <tr> <td>0000_B</td> <td>TAUD1 channel 0</td> </tr> <tr> <td>0001_B</td> <td>TAUD1 channel 1</td> </tr> <tr> <td>0010_B</td> <td>TAUD1 channel 2</td> </tr> <tr> <td>:</td> <td></td> </tr> <tr> <td>1101_B</td> <td>TAUD1 channel 13</td> </tr> <tr> <td>1110_B</td> <td>TAUD1 channel 14</td> </tr> <tr> <td>1111_B</td> <td>TAUD1 channel 15</td> </tr> </tbody> </table> | IC0CKSEL01[3:0] | Selected TAUD1 Channel | 0000 _B | TAUD1 channel 0 | 0001 _B | TAUD1 channel 1 | 0010 _B | TAUD1 channel 2 | : | | 1101 _B | TAUD1 channel 13 | 1110 _B | TAUD1 channel 14 | 1111 _B | TAUD1 channel 15 |
| IC0CKSEL01[3:0] | Selected TAUD1 Channel | | | | | | | | | | | | | | | | | |
| 0000 _B | TAUD1 channel 0 | | | | | | | | | | | | | | | | | |
| 0001 _B | TAUD1 channel 1 | | | | | | | | | | | | | | | | | |
| 0010 _B | TAUD1 channel 2 | | | | | | | | | | | | | | | | | |
| : | | | | | | | | | | | | | | | | | | |
| 1101 _B | TAUD1 channel 13 | | | | | | | | | | | | | | | | | |
| 1110 _B | TAUD1 channel 14 | | | | | | | | | | | | | | | | | |
| 1111 _B | TAUD1 channel 15 | | | | | | | | | | | | | | | | | |
| 3 to 0 | IC0CKSEL00 [3:0] | Select the counter-clock-enable signal for the OSTM0 counter (Enabled only when IC0TMEN0 = 1 and IC0TMSEL0[1:0] = 00 _B). | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0CKSEL00[3:0]</th> <th>Selected TAUD0 Channel</th> </tr> </thead> <tbody> <tr> <td>0000_B</td> <td>TAUD0 channel 0</td> </tr> <tr> <td>0001_B</td> <td>TAUD0 channel 1</td> </tr> <tr> <td>0010_B</td> <td>TAUD0 channel 2</td> </tr> <tr> <td>:</td> <td></td> </tr> <tr> <td>1101_B</td> <td>TAUD0 channel 13</td> </tr> <tr> <td>1110_B</td> <td>TAUD0 channel 14</td> </tr> <tr> <td>1111_B</td> <td>TAUD0 channel 15</td> </tr> </tbody> </table> | IC0CKSEL00[3:0] | Selected TAUD0 Channel | 0000 _B | TAUD0 channel 0 | 0001 _B | TAUD0 channel 1 | 0010 _B | TAUD0 channel 2 | : | | 1101 _B | TAUD0 channel 13 | 1110 _B | TAUD0 channel 14 | 1111 _B | TAUD0 channel 15 |
| IC0CKSEL00[3:0] | Selected TAUD0 Channel | | | | | | | | | | | | | | | | | |
| 0000 _B | TAUD0 channel 0 | | | | | | | | | | | | | | | | | |
| 0001 _B | TAUD0 channel 1 | | | | | | | | | | | | | | | | | |
| 0010 _B | TAUD0 channel 2 | | | | | | | | | | | | | | | | | |
| : | | | | | | | | | | | | | | | | | | |
| 1101 _B | TAUD0 channel 13 | | | | | | | | | | | | | | | | | |
| 1110 _B | TAUD0 channel 14 | | | | | | | | | | | | | | | | | |
| 1111 _B | TAUD0 channel 15 | | | | | | | | | | | | | | | | | |

22.3.2.10 IC0CKSEL1 — OSTM1 Clock Select Register

This register selects the counter-clock-enable signal as the clock source to drive counting for the OSTM1 counter.

Access: This register can be read/written in 16-bit units.

Address: FFDD 6004_H

Value after reset: 0000_H. This register is initialized by a reset of any type.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------|----|-----------------|------------------|------------------|------------------|-----|-----|------------------|-----|-----|-----|-----|-----|-----|-----|
| | IC0TMEN1 | — | IC0TMSEL1 [1:0] | IC0CKSEL13 [1:0] | IC0CKSEL12 [1:0] | IC0CKSEL11 [3:0] | | | IC0CKSEL10 [3:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 22.18 IC0CKSEL1 Register Contents (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | |
|-------------------|------------------------------|---|-------------------|------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 15 | IC0TMEN1 | Selects the counter-clock-enable signal for OSTM1. 0: The counter-clock-enable signal is fixed to 1 (PCLK is selected as the source of the clock signal for counting) 1: Selects the peripheral module selected by IC0TMSEL1[1:0] as the counter-clock-enable signal. | | | | | | | | | | |
| 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | |
| 13, 12 | IC0TMSEL1 [1:0] | Select the counter-clock-enable signal for the OSTM1 counter (Enabled only when IC0TMEN1 = 1). | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0TMSEL1[1:0]</th> <th>Selected Peripheral Function</th> </tr> </thead> <tbody> <tr> <td>00_B</td> <td>TAUD0</td> </tr> <tr> <td>01_B</td> <td>TAUD1</td> </tr> <tr> <td>10_B</td> <td>TAUJ0</td> </tr> <tr> <td>11_B</td> <td>TAUJ1</td> </tr> </tbody> </table> | IC0TMSEL1[1:0] | Selected Peripheral Function | 00 _B | TAUD0 | 01 _B | TAUD1 | 10 _B | TAUJ0 | 11 _B | TAUJ1 |
| IC0TMSEL1[1:0] | Selected Peripheral Function | | | | | | | | | | | |
| 00 _B | TAUD0 | | | | | | | | | | | |
| 01 _B | TAUD1 | | | | | | | | | | | |
| 10 _B | TAUJ0 | | | | | | | | | | | |
| 11 _B | TAUJ1 | | | | | | | | | | | |
| 11, 10 | IC0CKSEL13 [1:0] | Select the counter-clock-enable signal for the OSTM1 counter (Enabled only when IC0TMEN1 = 1 and IC0TMSEL1[1:0] = 11 _B). | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0CKSEL13[1:0]</th> <th>Selected TAUJ1 Channel</th> </tr> </thead> <tbody> <tr> <td>00_B</td> <td>TAUJ1 channel 0</td> </tr> <tr> <td>01_B</td> <td>TAUJ1 channel 1</td> </tr> <tr> <td>10_B</td> <td>TAUJ1 channel 2</td> </tr> <tr> <td>11_B</td> <td>TAUJ1 channel 3</td> </tr> </tbody> </table> | IC0CKSEL13[1:0] | Selected TAUJ1 Channel | 00 _B | TAUJ1 channel 0 | 01 _B | TAUJ1 channel 1 | 10 _B | TAUJ1 channel 2 | 11 _B | TAUJ1 channel 3 |
| IC0CKSEL13[1:0] | Selected TAUJ1 Channel | | | | | | | | | | | |
| 00 _B | TAUJ1 channel 0 | | | | | | | | | | | |
| 01 _B | TAUJ1 channel 1 | | | | | | | | | | | |
| 10 _B | TAUJ1 channel 2 | | | | | | | | | | | |
| 11 _B | TAUJ1 channel 3 | | | | | | | | | | | |
| 9, 8 | IC0CKSEL12 [1:0] | Select the counter-clock-enable signal for the OSTM1 counter (Enabled only when IC0TMEN1 = 1 and IC0TMSEL1[1:0] = 10 _B). | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>IC0CKSEL[121:120]</th> <th>Selected TAUJ0 Channel</th> </tr> </thead> <tbody> <tr> <td>00_B</td> <td>TAUJ0 channel 0</td> </tr> <tr> <td>01_B</td> <td>TAUJ0 channel 1</td> </tr> <tr> <td>10_B</td> <td>TAUJ0 channel 2</td> </tr> <tr> <td>11_B</td> <td>TAUJ0 channel 3</td> </tr> </tbody> </table> | IC0CKSEL[121:120] | Selected TAUJ0 Channel | 00 _B | TAUJ0 channel 0 | 01 _B | TAUJ0 channel 1 | 10 _B | TAUJ0 channel 2 | 11 _B | TAUJ0 channel 3 |
| IC0CKSEL[121:120] | Selected TAUJ0 Channel | | | | | | | | | | | |
| 00 _B | TAUJ0 channel 0 | | | | | | | | | | | |
| 01 _B | TAUJ0 channel 1 | | | | | | | | | | | |
| 10 _B | TAUJ0 channel 2 | | | | | | | | | | | |
| 11 _B | TAUJ0 channel 3 | | | | | | | | | | | |

Table 22.18 IC0CKSEL1 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|---------------------|--|
| 7 to 4 | IC0CKSEL11 [3:0] | Select the counter-clock-enable signal for the OSTM1 counter (Enabled only when IC0TMEN1 = 1 and IC0TMSEL1[1:0] = 01 _B). |
| | | IC0CKSEL11[3:0] Selected TAUD1 Channel |
| | | 0000 _B TAUD1 channel 0 |
| | | 0001 _B TAUD1 channel 1 |
| | | 0010 _B TAUD1 channel 2 |
| | | : |
| | | 1101 _B TAUD1 channel 13 |
| | | 1110 _B TAUD1 channel 14 |
| | | 1111 _B TAUD1 channel 15 |
| 3 to 0 | IC0CKSEL10 [3:0] | Select the counter-clock-enable signal for the OSTM1 counter (Enabled only when IC0TMEN1 = 1 and IC0TMSEL1[1:0] = 00 _B). |
| | | IC0CKSEL10[3:0] Selected TAUD0 Channel |
| | | 0000 _B TAUD0 channel 0 |
| | | 0001 _B TAUD0 channel 1 |
| | | 0010 _B TAUD0 channel 2 |
| | | : |
| | | 1101 _B TAUD0 channel 13 |
| | | 1110 _B TAUD0 channel 14 |
| | | 1111 _B TAUD0 channel 15 |

22.4 Operation

22.4.1 Starting and Stopping OSTMn

The OSTMn is started and stopped as follows:

Starting the timer

The timer is started in either of the following ways:

- setting the OSTMnTS.OSTMnTS bit to 1 or
- transitions of the OSTMnTSST signal from 0 to 1

The OSTMnTE.OSTMnTE status flag is set to 1.

The counter starts to count up or down in accord with the settings for operating mode. For details, refer to **Section 22.4.2, Interval Timer Mode** and **Section 22.4.3, Free-Running Comparison Mode**.

If the OSTMnTS.OSTMnTS bit is to be used to start the timer, the OSTMnTSST input must correspond to logical zero.

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops the OSTMn.

This also clears the OSTMnTE.OSTMnTE status flag.

When the counter is stopped, the OSTMnTO register and the OSTMnCNT register retain their current values and the output signal OSTMnTTOUT stays at the same level until the next time counting is started.

Synchronous start

The OSTMnTSST signal can be used to start the timer in synchronization with other peripheral functions. Refer to **Section 29.2.3.1, Simultaneous Start Trigger Function**.

22.4.2 Interval Timer Mode

In interval timer mode, the OSTMn can be used as a reference timer generating interrupt requests at fixed intervals.

22.4.2.1 Basic Operation in Interval Timer Mode

In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An INTOSTMn interrupt request is generated when the counter underflows (reaches 0000 0000_H).

To select interval timer mode, set OSTMnCTL.OSTMnMD1 = 0.

New values can be written to the OSTMnCMP register at any time. When a new value is written while counting is in progress, the new OSTMnCMP value will only be loaded the next time the counter reaches 0000 0000_H. The counter then continues with the new value.

Intervals for INTOSTMn and OSTMnTTOUT output

The intervals for INTOSTMn and OSTMnTTOUT output are as follows.

- INTOSTMn generation cycle = counter-clock cycle × (OSTMnCMP + 1)
- OSTMnTTOUT output cycle = INTOSTMn generation cycle × 2

The following figure shows the basic operation of the OSTMn in interval timer mode with counter-start interrupts enabled.

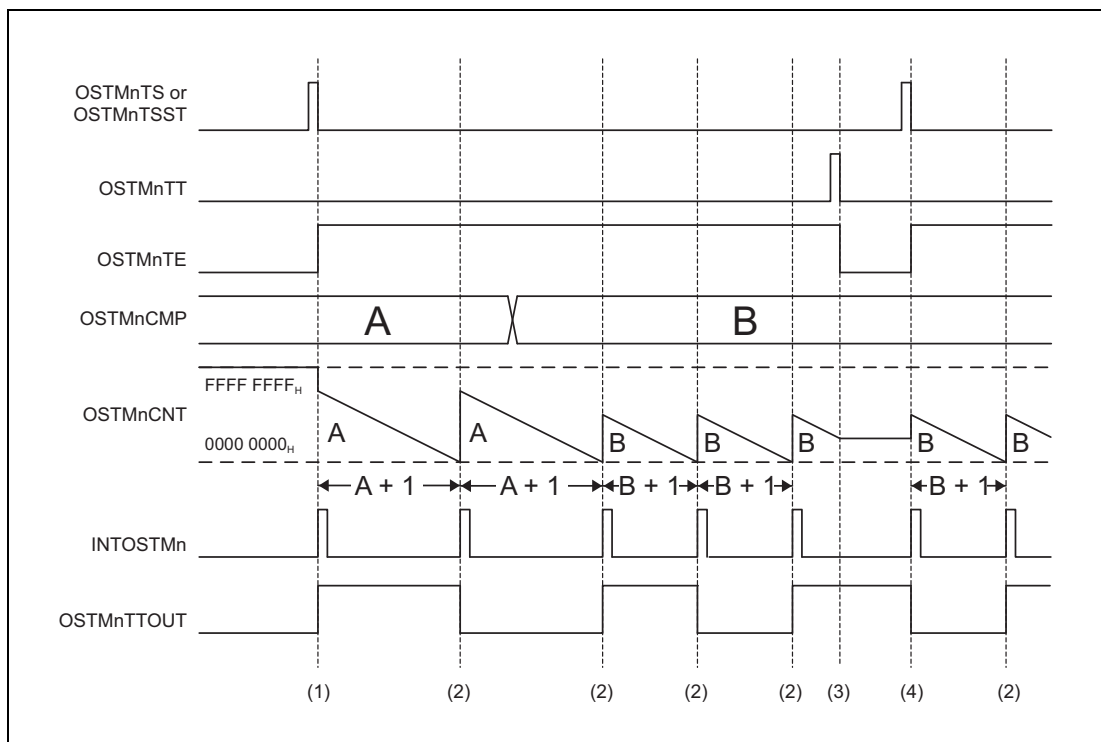


Figure 22.7 Timing Diagram of OSTMn in Interval Timer Mode

The timing diagram above shows the following:

- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter starts counting-down from the value of OSTMnCMP. If OSTMnCTL.OSTMnMD0 is 1, INTOSTMn interrupt requests are generated at the start of counting and the OSTMnTTOUT signal is toggled. The OSTMnCNT register indicates the counter value.
- (2) When the counter reaches 0000 0000_H, an INTOSTMn interrupt request is generated and the OSTMnTTOUT signal is toggled. The counter loads the new start value from OSTMnCMP and continues counting down.
- (3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
- (4) When counting is restarted (OSTMnTS.OSTMnTS = 1, or OSTMnTSST = 1), the counter loads the new start value from OSTMnCMP and starts counting down.

Forced restart

The counter is forcibly restarted by setting $OSTMnTS.OSTMnTS = 1$ or by a transition of the $OSTMnTSST$ signal from 0 to 1 during counting.

The counter loads the start value from the $OSTMnCMP$ register and continues counting down.

The following figure shows a timing diagram in interval timer mode, with counter-start interrupts enabled ($OSTMnCTL.OSTMnMD0 = 1$), and in timer output toggling mode, where $OSTMnTTOUT$ is toggled ($OSTMnTOE.OSTMnTOE = 1$).

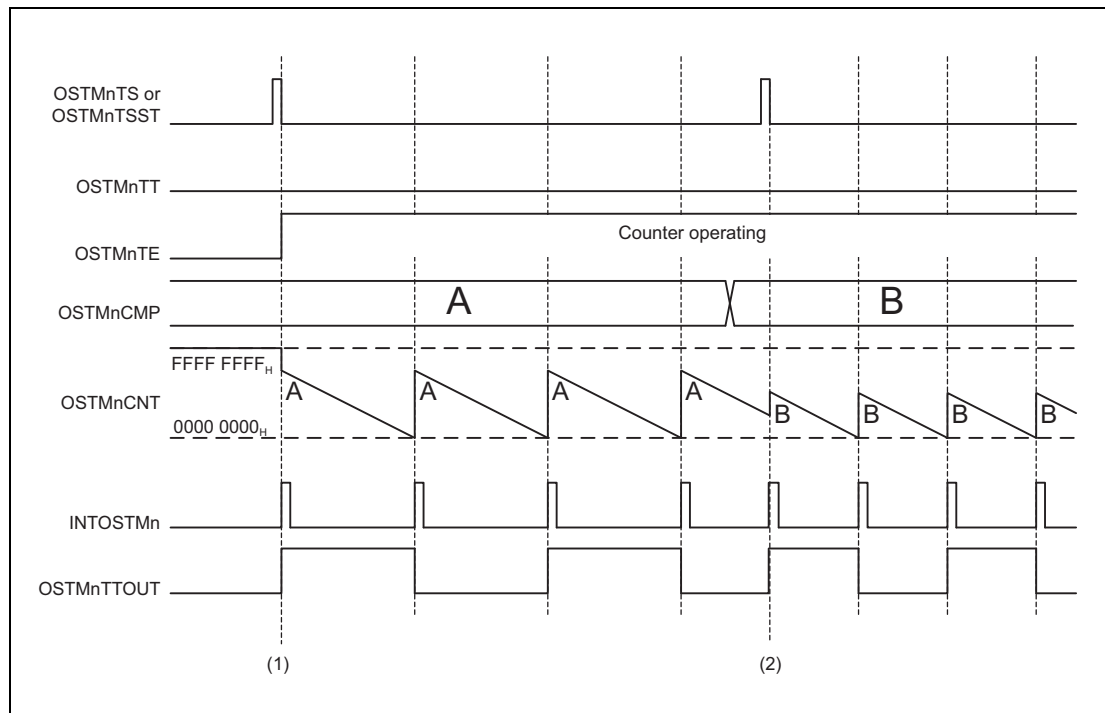


Figure 22.8 Timing Diagram of Forced Restart in Interval Timer Mode

The timing diagram above shows the following operations:

- (1) The counter is started and stopped as described under **Figure 22.7, Timing Diagram of OSTMn in Interval Timer Mode**.
- (2) Setting $OSTMnTS.OSTMnTS = 1$ or $OSTMnTSST = 1$ restarts the counter while counting is in progress (i.e. while $OSTMnTE.OSTMnTE = 1$). The counter immediately restarts counting down, starting with the current value of $OSTMnCMP$. When $OSTMnCTL.OSTMnMD0 = 1$, an $INTOSTMn$ interrupt request is generated when counting starts and the $OSTMnTTOUT$ signal is toggled.

22.4.2.2 Operation when $OSTMnCMP = 0000\ 0000_H$

When PCLK is selected as the source of the clock signal for counting and $OSTMnCMP = 0000\ 0000_H$, the OSTMn behaves as follows.

- If the counter is enabled, the INTOSTMn interrupt request will always be set to 1. The INTOSTMn interrupt request is generated only when the counter starts counting. However, the timer (OSTMnTTOUT) output signal can still be used. Timer (OSTMnTTOUT) output using timer output toggling mode results in OSTMnTTOUT being toggled on every cycle of the counter clock.

The following figure shows operations of the OSTMn when PCLK is selected as the source of the clock signal for counting and $OSTMnCMP = 0000\ 0000_H$, the counter start interrupt is enabled ($OSTMnCTL.OSTMnMD0 = 1$), and the timer is in timer output toggling mode, where OSTMnTTOUT is toggled ($OSTMnTOE.OSTMnTOE = 1$).

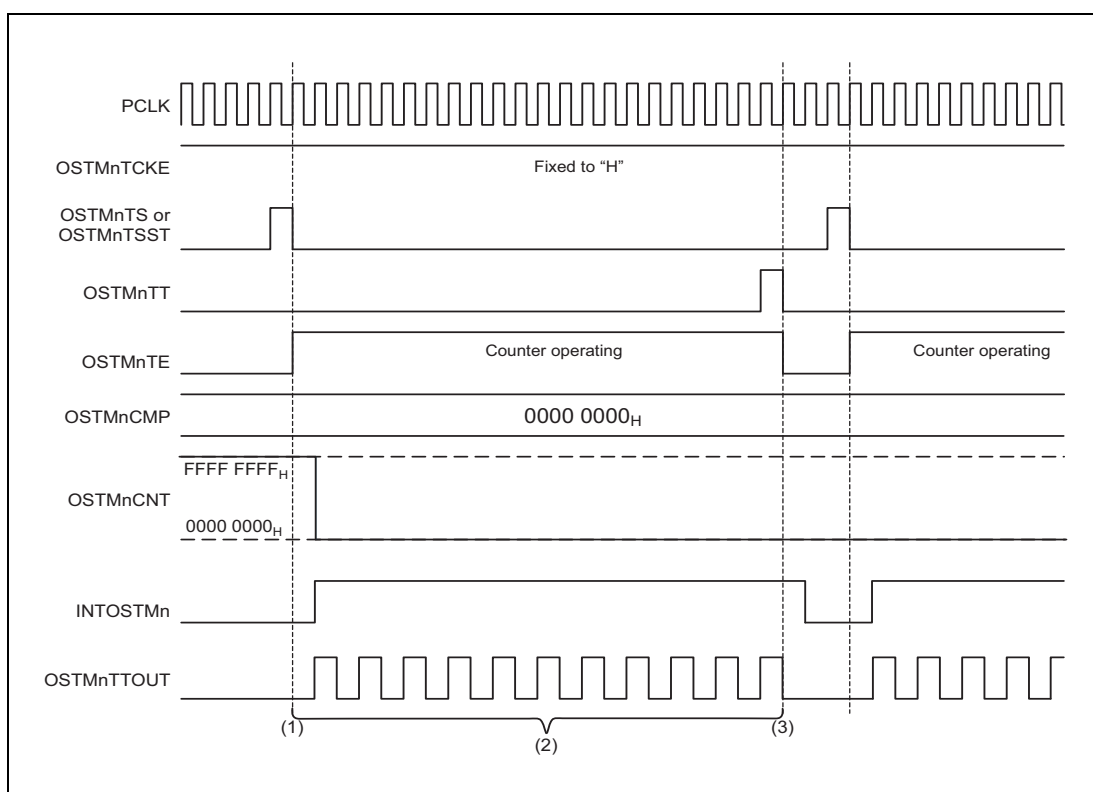


Figure 22.9 Timing Diagram when Counter Clock = PCLK (OSTMnTCKE is at High Level) and $OSTMnCMP = 0000\ 0000_H$ in Interval Timer Mode

The timing diagram above shows the following operations:

- (1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value $0000\ 0000_H$ is retained in OSTMnCMP.
- (2) While the INTOSTMn interrupt request signal is continuously asserted, OSTMnTTOUT is toggled (in **Figure 22.9**, INTOSTMn is fixed to the high level because PCLK is selected as the counter clock).
- (3) After the counter stops, the INTOSTMn interrupt request signal is deasserted and the output level of the OSTMnTTOUT signal is maintained.

When an interrupt is prohibited when the counter starts, 1 clock cycle of the counter clock is not generated at the start timing of the counter.

22.4.2.3 Setting Procedure for Interval Timer Mode

The procedure for setting up interval timer mode after release from the reset state is described below.

Setting procedure

- (1) Set the start value of the counter in the OSTMnCMP register.
- (2) To output OSTMnTTOUT:
 - Initialize the OSTMnTO register in software control mode (OSTMnTOE.OSTMnTOE = 0).
 - Select timer output toggle mode (OSTMnTOE.OSTMnTOE = 1).
- (3) Select interval timer mode by setting the OSTMnCTL.OSTMnMD1 bit to 0.
- (4) Select by using the OSTMnCTL.OSTMnMD0 bit whether to enable or disable an interrupt when counting starts.

22.4.3 Free-Running Comparison Mode

22.4.3.1 Basic Operation in Free-Running Comparison Mode

In free-running comparison mode, the counter counts up from $0000\ 0000_H$ to $FFFF\ FFFF_H$. When the value of the OSTMnCMP register matches the current counter value, an INTOSTMn interrupt request is generated.

When the free-running comparison mode is used, set the OSTMnCTL.OSTMnMD1 to 1.

New values can be written to the OSTMnCMP register at any time.

The following figure shows the basic operation of the OSTMn in free-running compare mode with the start of counting enabled (OSTMnCTL.OSTMnMD0 = 1) and with the timer in timer output toggling mode, where OSTMnTTOUT is toggled (OSTMnTOE.OSTMnTOE = 1).

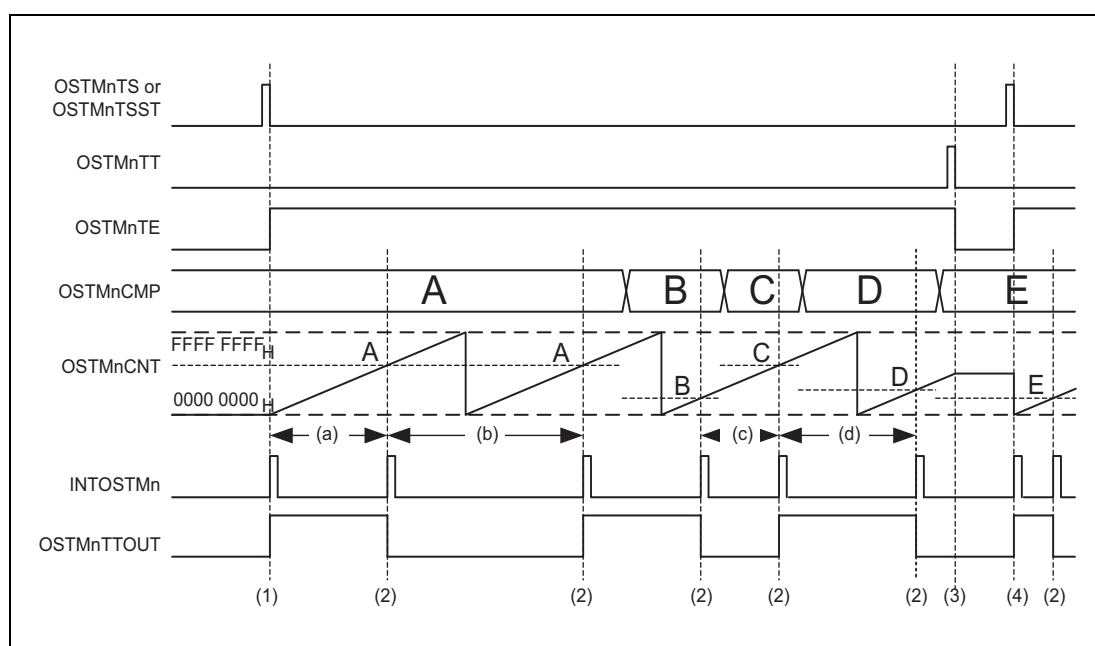


Figure 22.10 Timing Diagram of OSTMn in Free-Running Comparison Mode

The timing diagram above shows the following:

- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from $0000\ 0000_H$ to $FFFF\ FFFF_H$. The OSTMnCNT register indicates the counter value. When the OSTMnCTL.OSTMnMD0 set to 1, the interrupt request INTOSTMn is generated when the counting starts.
- (2) When the current counter value matches the value in the OSTMnCMP register, an INTOSTMn interrupt request is generated and the OSTMnTTOUT signal is toggled.
- (3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
- (4) Counting by the counter restarts from $0000\ 0000_H$ when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST = 1.

INTOSTMn Period

The INTOSTMn generation interval depends on whether or not it is the first time after start counting or not, and if OSTTMnCMP is rewritten during operation, the INTOSTMn generation interval also depends on the counter value at timing of rewriting, the old comparison value and the new comparison value.

Table 22.19 Interval of INTOSTMn Generation

| Old value for comparison | New value for comparison | Counter value at time of rewriting | Interval to generate the next INTOSTMn | Label in timing diagram |
|--------------------------|--------------------------|------------------------------------|--|-------------------------|
| | | Counter starts | $(A + 1) \times$ counter clock period | (a) |
| A | A | No rewriting | $(FFFF\ FFFF_H + 1) \times$ counter clock period | (b) |
| B | $C > B$ | $B < \text{counter value} < C$ | $(C - B) \times$ counter clock period | (c) |
| C | $D < C$ | Counter value $> D, C$ | $(FFFF\ FFFF_H - C + D + 1) \times$ counter clock period | (d) |

Forced restart

Forced restart is not performed when the OSTMnTS.OSTMnTS bit is set or OSTMnTSST = 1 during counting. The counter ignores the attempted setting and continues counting.

22.4.3.2 Operation when $OSTMnCMP = 0000\ 0000_H$

The following figure shows the operation of the OSTMn when PCLK is selected as the source of the clock signal for counting and $OSTMnCMP = 0000\ 0000_H$, counter-start interrupts are enabled ($OSTMnCTL.OSTMnMD0 = 1$) and the timer is in timer output toggling mode, where $OSTMnTTOUT$ is toggled ($OSTMnTOE.OSTMnTOE = 1$).

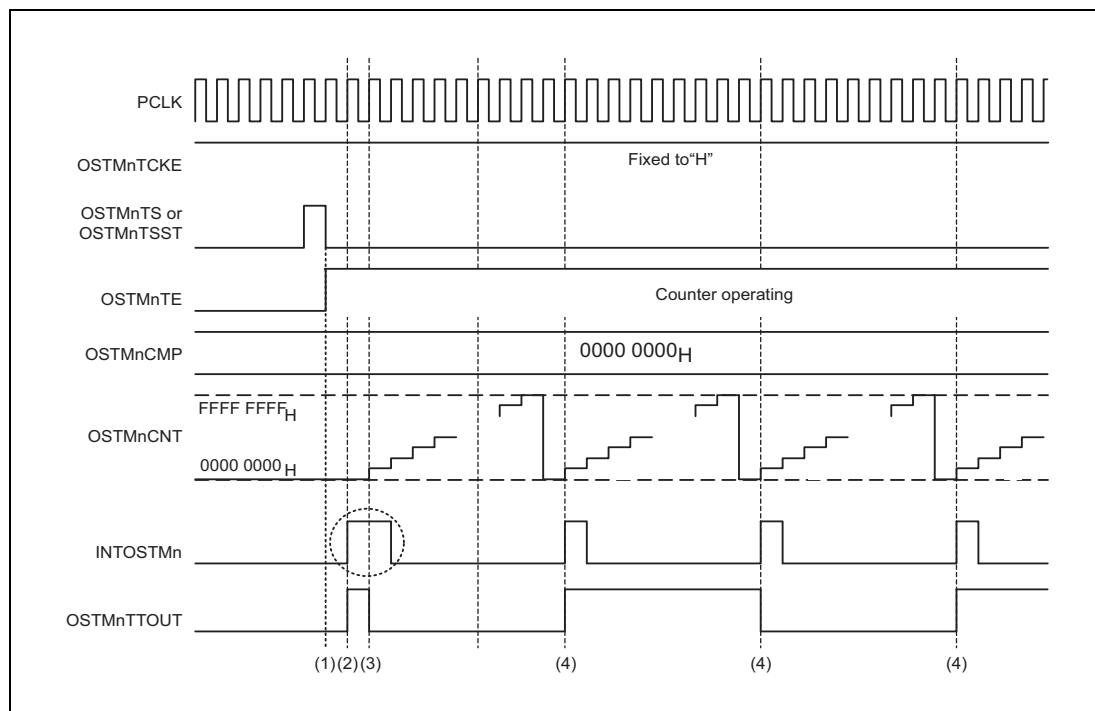


Figure 22.11 Timing Diagram when Counter Clock = PCLK (OSTMnTCKE is at High Level) and $OSTMnCMP = 0000\ 0000_H$ in Free-Running Comparison Mode

The timing diagram above shows the following operations.

- (1) Once the counter starts, it counts up from $0000\ 0000_H$ to $FFFF\ FFFF_H$.
- (2) An INTOSTMn interrupt request is generated when counting starts and the OSTMnTTOUT signal is toggled. Though INTOSTMn is at the high level for the period of two cycles of PCLK at the start of counting operations, only a single interrupt request is generated on the rising edge of INTOSTMn.
- (3) If the current counter value matches OSTMnCMP, the interrupt request INTOSTMn is generated. If $OSTMnCMP = 0000\ 0000_H$ as shown above, INTOSTMn is generated over two clock cycles and the OSTMnTTOUT signal is toggled.
- (4) For every $(FFFF\ FFFF_H + 1)$ clock cycles the INTOSTMn interrupt request signal is asserted and the OSTMnTTOUT signal is toggled.

When an interrupt is prohibited when the counter starts, 1 clock cycle of the counter clock is not generated at the start timing of the counter.

22.4.3.3 Setting Procedure for Free-Running Comparison Mode

The setting procedure in free-running compare mode after reset release is described below:

Initialization

- (1) Set the comparison value in the OSTMnCMP register.
- (2) To output OSTMnTTOUT:
 - Initialize the OSTMnTO register in software control mode (OSTMnTOE.OSTMnTOE = 0).
 - Select timer output toggle mode (OSTMnTOE.OSTMnTOE = 1).
- (3) Select free-running compare mode by setting the OSTMnCTL.OSTMnMD1 bit to 1.
- (4) Enable or disable an interrupt when counting is started by the OSTMnCTL.OSTMnMD0 bit.

Section 23 Timer Array Unit D (TAUD)

The Timer Array Unit D (TAUD) is multifunction timer unit with several 16-bit timers (16-bit timer array unit). TAUD is implemented 3 units with the channel interface of 16 per one unit.

23.1 Features of RH850/P1M-E TAUD

23.1.1 Units and Channels

This microcontroller has the following number of TAUD units.

Table 23.1 Number of Units

| Product | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 3 | 3 |
| Name | TAUDn (n = 0 to 2) | TAUDn (n = 0 to 2) |

TAUDn has the following timers for the quantity of channels of timers.

Table 23.2 TAUDn Unit Configurations and Channels

| Unit Name (Channel Name) | Channels per Unit | RH850/P1M-E 100 pins (48 ch) | RH850/P1M-E 144 pins (48 ch) |
|-----------------------------|-------------------|------------------------------------|------------------------------------|
| TAUD0 | 16 | √ | √ |
| TAUD1 | 16 | √ | √ |
| TAUD2 | 16 | √ | √ |

Table 23.3 Index

| Index | Description |
|-------|---|
| n | Throughout this section, the individual TAUD units are identified by the index "n" (n = 0 to 2) |
| m | The TAUD has 16 channels. Throughout this section, the individual channels are identified by the index "m" (m = 0 to 15), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd. |

23.1.2 Register Base Address

TAUDn base addresses are listed in the following table.

TAUDn register addresses are given as offsets from the base addresses throughout the section.

Table 23.4 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <TAUD0_base> | FFE2 0000 _H |
| <TAUD1_base> | FFE2 1000 _H |
| <TAUD2_base> | FFE2 2000 _H |

23.1.3 Clock Supply

Clock supply by and to TAUDn is listed in the following table.

Table 23.5 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|-------------------------------------|
| TAUDn | PCLK | High-speed peripheral clock CLK_HSB |

23.1.4 Interrupt Requests

TAUDn interrupt requests are listed in the following table.

Table 23.6 Interrupt Requests (1/2)

| Unit Interrupt Name | Description | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|----------------------|------------------|------------------------|
| TAUD0 | | | |
| INTTAUD0I0 | Channel 0 interrupt | 141 | 13 |
| INTTAUD0I1 | Channel 1 interrupt | 142 | 14 |
| INTTAUD0I2 | Channel 2 interrupt | 143 | 15 |
| INTTAUD0I3 | Channel 3 interrupt | 144 | 16 |
| INTTAUD0I4 | Channel 4 interrupt | 145 | 17 |
| INTTAUD0I5 | Channel 5 interrupt | 146 | 18 |
| INTTAUD0I6 | Channel 6 interrupt | 147 | 19 |
| INTTAUD0I7 | Channel 7 interrupt | 148 | 20 |
| INTTAUD0I8 | Channel 8 interrupt | 149 | 21 |
| INTTAUD0I9 | Channel 9 interrupt | 150 | 22 |
| INTTAUD0I10 | Channel 10 interrupt | 151 | 23 |
| INTTAUD0I11 | Channel 11 interrupt | 152 | 24 |
| INTTAUD0I12 | Channel 12 interrupt | 153 | 25 |
| INTTAUD0I13 | Channel 13 interrupt | 154 | 26 |
| INTTAUD0I14 | Channel 14 interrupt | 155 | 27 |
| INTTAUD0I15 | Channel 15 interrupt | 156 | 28 |
| TAUD1 | | | |
| INTTAUD1I0 | Channel 0 interrupt | 158 | 29 |
| INTTAUD1I1 | Channel 1 interrupt | 159 | 30 |

Table 23.6 Interrupt Requests (2/2)

| Unit Interrupt Name | Description | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|----------------------|------------------|------------------------|
| INTTAUD1I2 | Channel 2 interrupt | 160 | 31 |
| INTTAUD1I3 | Channel 3 interrupt | 161 | 32 |
| INTTAUD1I4 | Channel 4 interrupt | 162 | 33 |
| INTTAUD1I5 | Channel 5 interrupt | 163 | 34 |
| INTTAUD1I6 | Channel 6 interrupt | 164 | 35 |
| INTTAUD1I7 | Channel 7 interrupt | 165 | 36 |
| INTTAUD1I8 | Channel 8 interrupt | 166 | 37 |
| INTTAUD1I9 | Channel 9 interrupt | 167 | 38 |
| INTTAUD1I10 | Channel 10 interrupt | 168 | 39 |
| INTTAUD1I11 | Channel 11 interrupt | 169 | 40 |
| INTTAUD1I12 | Channel 12 interrupt | 170 | 41 |
| INTTAUD1I13 | Channel 13 interrupt | 171 | 42 |
| INTTAUD1I14 | Channel 14 interrupt | 172 | 43 |
| INTTAUD1I15 | Channel 15 interrupt | 173 | 44 |
| TAUD2 | | | |
| INTTAUD2I0 | Channel 0 interrupt | 260 | 45 |
| INTTAUD2I1 | Channel 1 interrupt | 261 | 46 |
| INTTAUD2I2 | Channel 2 interrupt | 262 | 47 |
| INTTAUD2I3 | Channel 3 interrupt | 263 | 48 |
| INTTAUD2I4 | Channel 4 interrupt | 264 | 49 |
| INTTAUD2I5 | Channel 5 interrupt | 265 | 50 |
| INTTAUD2I6 | Channel 6 interrupt | 266 | 51 |
| INTTAUD2I7 | Channel 7 interrupt | 267 | 52 |
| INTTAUD2I8 | Channel 8 interrupt | 268 | 53 |
| INTTAUD2I9 | Channel 9 interrupt | 269 | 54 |
| INTTAUD2I10 | Channel 10 interrupt | 270 | — |
| INTTAUD2I11 | Channel 11 interrupt | 271 | — |
| INTTAUD2I12 | Channel 12 interrupt | 272 | — |
| INTTAUD2I13 | Channel 13 interrupt | 273 | — |
| INTTAUD2I14 | Channel 14 interrupt | 274 | — |
| INTTAUD2I15 | Channel 15 interrupt | 275 | — |

23.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

23.1.6 External Input/Output Signals

External input/output signals of TAUDn are listed below.

Table 23.7 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|---|-----|-----------------------------|-----------------------------------|
| TAUD0 | | | |
| TAUD0TTIN0 to TAUD0TTIN15* ² | I | Inputs for channel 0 to 15 | TAUD0I0 to TAUD0I15 |
| TAUD0TTOUT0 to TAUD0TTOUT15 | O | Outputs for channel 0 to 15 | TAUD0O0 to TAUD0O15 |
| TAUD1 | | | |
| TAUD1TTIN0 to TAUD1TTIN15* ² | I | Inputs for channel 0 to 15 | TAUD1I0 to TAUD1I15 |
| TAUD1TTOUT0 to TAUD1TTOUT15 | O | Outputs for channel 0 to 15 | TAUD1O0 to TAUD1O15 |
| TAUD2 | | | |
| TAUD2TTIN0 to TAUD2TTIN15* ² | I | Inputs for channel 0 to 15 | TAUD2I0 to TAUD2I15* ¹ |
| TAUD2TTOUT0 to TAUD2TTOUT15 | O | Outputs for channel 0 to 15 | TAUD2O0 to TAUD2O15 |

Note 1. The input signals can be switched by PIC. For details, see **Section 29.2.3.11, TAUD Input Select Function**, and **Section 29.2.3.13, Timer Output Monitor Function (PWM-Diag)**.

Note 2. Setting of the noise filter for the port is required when the channel input pin is used. For details, **Section 2.6, Noise Filter and Edge Level Detection Circuit**.

23.1.7 Internal Input/Output Signals

The internal input/output signals of TAUDn are listed below.

Table 23.8 Internal Input/Output Signals

| Unit Signal Name | Description | Connected to |
|--------------------------|--|--------------|
| TAUDnTSSTm* ¹ | Simultaneous channel start trigger input | PIC |

Note 1. n = 0, 1 only. TAUD2TSSTm is not connected to PIC.

23.2 Overview

23.2.1 Functional Overview

The TAUD has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

The Timer Array Unit D is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16 bit counter TAUDnCNTm and a 16-bit data register TAUDnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in different operation modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, its operation mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, this means that one channel can control when another starts to count, is reset, etc.

23.2.2 Terms

In this chapter, the following terms are used:

Independent / synchronous channel operation

Independent or synchronous channel operation describes the dependency of channels on each other:

- If a channel operates independent of all other channels, this is called independent channel operation.
- If a channel operates depending on other channels, this is called synchronous channel operation.

Channel group

In synchronous channel operation, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Operation mode

An operation mode can be selected for every channel m . The operation mode defines the basic operation and features of a channel.

In synchronous channel operation, every channel in the channel group can operate in a different operation mode.

Examples are “Capture Mode”, “Event Count Mode”, and “Interval Timer Mode”.

Channel output mode

The channel output mode defines the operation of $TAUDnTTOUTm$

- of a single channel (independent output operation) or
- of all channels in a channel group (synchronous output operation).

Examples are “Independent Channel Output Mode 1” and “Synchronous Channel Output Mode 2 with Dead Time Output”.

Channel operation function

The channel operation function defines the complete function and all features

- of a single channel (independent channel operation) or
- of all channels in a channel group (synchronous channel operation).

Upper / lower channel

Depending on the channel number m , a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

Example:

For channel 5, channel 3 is an upper channel and channel 9 is a lower channel.

23.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 23.9 Functional List of TAUD Operations

| Function of Independent Operation | Example |
|--|-------------------------|
| Independent Channel Operation Functions | Section 23.12 |
| Interval Timer Function | Section 23.12.1 |
| TAUDnTTINm Input Interval Timer Function | Section 23.12.2 |
| Clock Divider Function | Section 23.12.3 |
| External Event Count Function | Section 23.12.4 |
| Delay Count Function | Section 23.12.5 |
| One-Pulse Output Function | Section 23.12.6 |
| TAUDnTTINm Input Pulse Interval Measurement Function | Section 23.12.7 |
| TAUDnTTINm Input Signal Width Measurement Function | Section 23.12.8 |
| TAUDnTTINm Input Position Detection Function | Section 23.12.9 |
| TAUDnTTINm Input Period Count Detection Function | Section 23.12.10 |
| TAUDnTTINm Input Pulse Interval Judgment Function | Section 23.12.11 |
| TAUDnTTINm Input Signal Width Judgment Function | Section 23.12.12 |
| One-Phase PWM Output Function | Section 23.12.13 |
| Real Time Output Function Type 1 | Section 23.12.14 |
| Real-Time Output Function Type 2 | Section 23.12.15 |
| Simultaneous Rewrite Trigger Generation Function Type 1 | Section 23.12.16 |
| Synchronous Channel Operation Functions | Section 23.13 |
| PWM Output Function | Section 23.13.1 |
| One-Shot Pulse Output Function | Section 23.13.2 |
| Delay Pulse Output Function | Section 23.13.3 |
| Offset Trigger Output Function | Section 23.13.4 |
| A/D Conversion Trigger Output Function Type 1 | Section 23.13.5 |
| Triangle PWM Output Function | Section 23.13.6 |
| Triangle PWM Output Function with Dead Time | Section 23.13.7 |
| A/D Conversion Trigger Output Function Type 2 | Section 23.13.8 |
| Interrupt Request Signals Culling Function | Section 23.13.9 |
| Synchronous Non-Complementary and Complementary Modulation Output Functions | Section 23.14 |
| Non-Complementary Modulation Output Function Type 1 | Section 23.14.1 |
| Non-Complementary Modulation Output Function Type 2 | Section 23.14.2 |
| Complementary Modulation Output Function | Section 23.14.3 |

23.2.4 TAUD I/O and Interrupt Request Signals

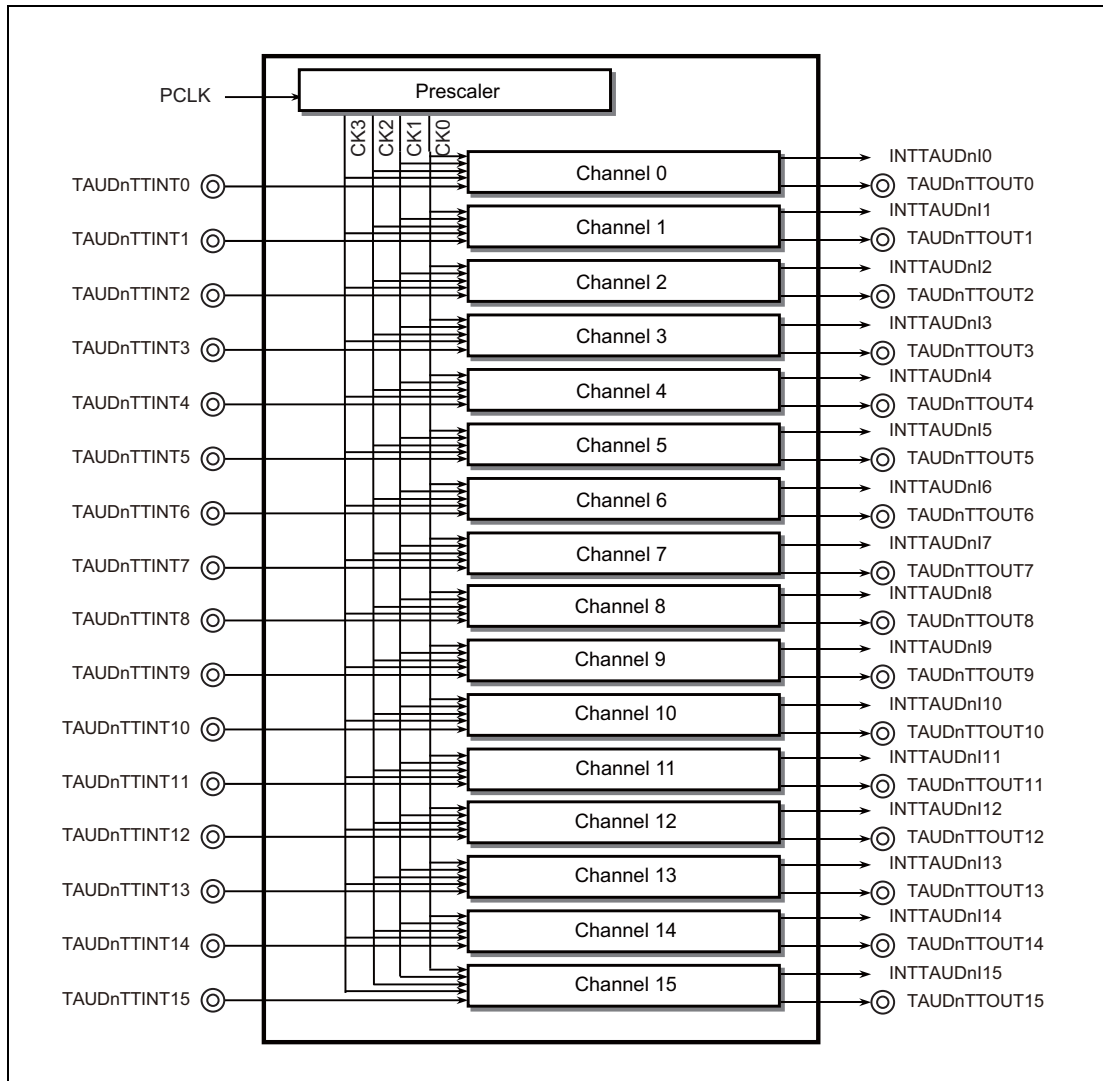


Figure 23.1 TAUD I/O and Interrupt Request Signals

23.2.5 Block Diagram

Figure 23.2 shows the main components of the TAUD:

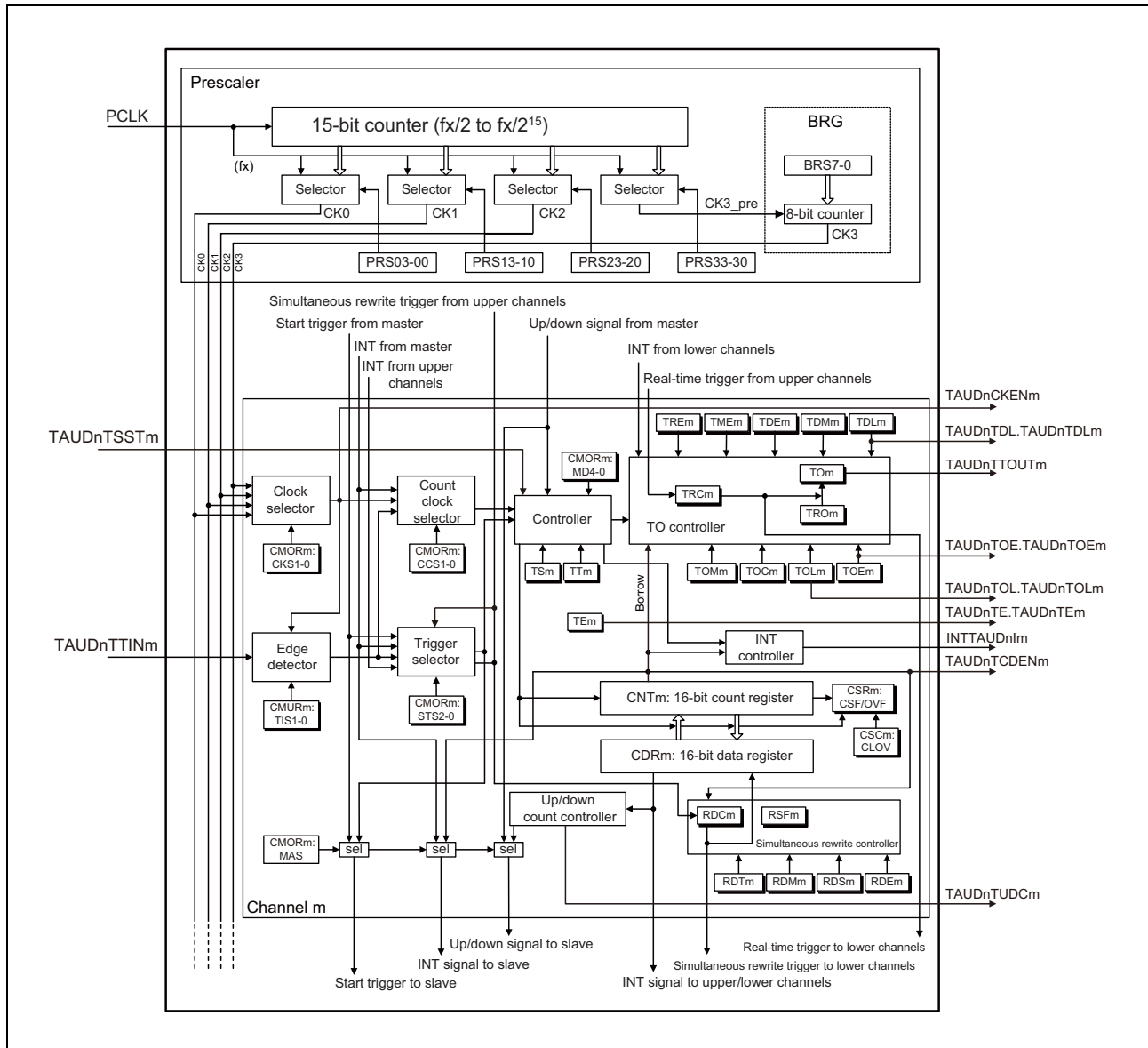


Figure 23.2 Block Diagram of the TAUD

The prefix “TAUDn” has been omitted from the module names for the sake of clarity in the above figure.

23.2.6 Description of Blocks

The following describes the functional blocks:

Prescaler

The prescaler provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived from PCLK by configurable prescaler division factors of 2^0 to 2^{15} . The fourth count clock CK3 can be adjusted more precisely by using the BRG to set an additional division factor that is not a power of 2.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- INTTAUDnIm from master channel
- TAUDnTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUDnCMORm.TAUDnMD[4:0])
- Counter start enable (TAUDnTS.TAUDnTSM) and counter stop (TAUDnTT.TAUDnTTm)

When counter start is enabled, status flag TAUDnTE.TAUDnTEm is set.

- Count direction (up/down) (can be controlled by master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUDnTE.TAUDnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- Synchronous channel start trigger input TAUDnTSSTm
- INTTAUDnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUDnTTOUTm generation
- TAUDnTTINm input signal valid edge

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operation modes. The data registers (TAUDnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUDnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

23.3 Registers

23.3.1 List of Registers

TAUDn registers are listed in the following table.

For details about <TAUDn_base>, see **Section 23.1.2, Register Base Address**.

Table 23.10 Registers

| Module | Register | Symbol | Address |
|------------------------------------|--|------------|--|
| TAUDn prescaler registers | | | |
| TAUDn | TAUDn prescaler clock select register | TAUDnTPS | <TAUDn_base> + 240 _H |
| TAUDn | TAUDn prescaler baud rate setting register | TAUDnBRS | <TAUDn_base> + 244 _H |
| TAUDn control registers | | | |
| TAUDn | TAUDn channel data register m | TAUDnCDRm | <TAUDn_base> + m × 4 _H |
| TAUDn | TAUDn channel counter register m | TAUDnCNTm | <TAUDn_base> + 80 _H + m × 4 _H |
| TAUDn | TAUDn channel mode OS register m | TAUDnCMORm | <TAUDn_base> + 200 _H + m × 4 _H |
| TAUDn | TAUDn channel mode user register m | TAUDnCMURm | <TAUDn_base> + C0 _H + m × 4 _H |
| TAUDn | TAUDn channel status register m | TAUDnCSRm | <TAUDn_base> + 140 _H + m × 4 _H |
| TAUDn | TAUDn channel status clear register m | TAUDnCSCm | <TAUDn_base> + 180 _H + m × 4 _H |
| TAUDn | TAUDn channel start trigger register | TAUDnTS | <TAUDn_base> + 1C4 _H |
| TAUDn | TAUDn channel enable status register | TAUDnTE | <TAUDn_base> + 1C0 _H |
| TAUDn | TAUDn channel stop trigger register | TAUDnTT | <TAUDn_base> + 1C8 _H |
| TAUDn output registers | | | |
| TAUDn | TAUDn channel output enable register | TAUDnTOE | <TAUDn_base> + 5C _H |
| TAUDn | TAUDn channel output register | TAUDnTO | <TAUDn_base> + 58 _H |
| TAUDn | TAUDn channel output mode register | TAUDnTOM | <TAUDn_base> + 248 _H |
| TAUDn | TAUDn channel output configuration register | TAUDnTOC | <TAUDn_base> + 24C _H |
| TAUDn | TAUDn channel output active level register | TAUDnTOL | <TAUDn_base> + 040 _H |
| TAUDn | TAUDn channel dead time output enable register | TAUDnTDE | <TAUDn_base> + 250 _H |
| TAUDn | TAUDn channel dead time output mode register | TAUDnTDM | <TAUDn_base> + 254 _H |
| TAUDn | TAUDn channel dead time output level register | TAUDnTDL | <TAUDn_base> + 54 _H |
| TAUDn | TAUDn channel real-time output register | TAUDnTRO | <TAUDn_base> + 4C _H |
| TAUDn | TAUDn channel real-time output enable register | TAUDnTRE | <TAUDn_base> + 258 _H |
| TAUDn | TAUDn channel real-time output control register | TAUDnTRC | <TAUDn_base> + 25C _H |
| TAUDn | TAUDn channel modulation output enable register | TAUDnTME | <TAUDn_base> + 50 _H |
| TAUDn reload data registers | | | |
| TAUDn | TAUDn channel reload data enable register | TAUDnRDE | <TAUDn_base> + 260 _H |
| TAUDn | TAUDn channel reload data mode register | TAUDnRDM | <TAUDn_base> + 264 _H |
| TAUDn | TAUDn channel reload data control CH select register | TAUDnRDS | <TAUDn_base> + 268 _H |
| TAUDn | TAUDn channel reload data control register | TAUDnRDC | <TAUDn_base> + 26C _H |
| TAUDn | TAUDn channel reload data trigger register | TAUDnRDT | <TAUDn_base> + 44 _H |
| TAUDn | TAUDn channel reload status register | TAUDnRSF | <TAUDn_base> + 48 _H |

23.3.2 Details of TAUDn Prescaler Registers

23.3.2.1 TAUDnTPS — TAUDn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescaler. CK3 is generated by dividing CK3_PRE by the factor specified in TAUDnBRS.

Access: Readable/writable in 16-bit units. Write to this register when TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 240_H

Value after reset: FFFF_H This register is initialized by any reset source.

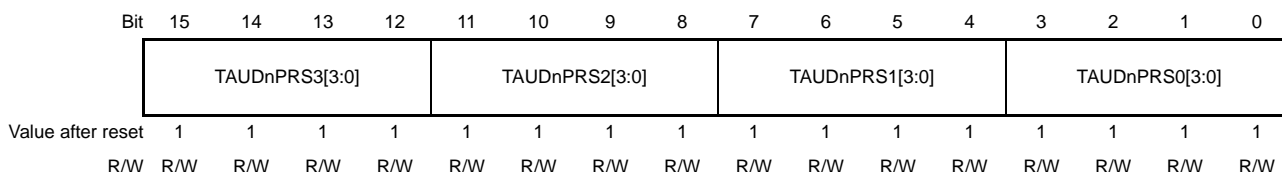


Table 23.11 TAUDnTPS Register Contents (1/3)

| Bit Position | Bit Name | Function |
|----------------|-------------------|---|
| 15 to 12 | TAUDnPRS3 [3:0] | Specify the CK3_PRE clock. CK3_PRE clock is an input clock to BRG unit which supplies the CK3 operating clock to all channels. |
| TAUDnPRS3[3:0] | | CK3_PRE Clock |
| | 0000 _B | PCLK/2 ⁰ |
| | 0001 _B | PCLK/2 ¹ |
| | 0010 _B | PCLK/2 ² |
| | 0011 _B | PCLK/2 ³ |
| | 0100 _B | PCLK/2 ⁴ |
| | 0101 _B | PCLK/2 ⁵ |
| | 0110 _B | PCLK/2 ⁶ |
| | 0111 _B | PCLK/2 ⁷ |
| | 1000 _B | PCLK/2 ⁸ |
| | 1001 _B | PCLK/2 ⁹ |
| | 1010 _B | PCLK/2 ¹⁰ |
| | 1011 _B | PCLK/2 ¹¹ |
| | 1100 _B | PCLK/2 ¹² |
| | 1101 _B | PCLK/2 ¹³ |
| | 1110 _B | PCLK/2 ¹⁴ |
| | 1111 _B | PCLK/2 ¹⁵ |

The above bits are rewritable only when all the counters using CK3 are stopped (TAUDnTE.TAUDnTEm = 0).

Table 23.11 TAUDnTPS Register Contents (2/3)

| Bit Position | Bit Name | Function | |
|--|----------------------|------------------------|----------------------|
| 11 to 8 | TAUDnPRS2 [3:0] | Specify the CK2 clock. | |
| | | TAUDnPRS2[3:0] | CK2 Clock |
| | | 0000 _B | PCLK/2 ⁰ |
| | | 0001 _B | PCLK/2 ¹ |
| | | 0010 _B | PCLK/2 ² |
| | | 0011 _B | PCLK/2 ³ |
| | | 0100 _B | PCLK/2 ⁴ |
| | | 0101 _B | PCLK/2 ⁵ |
| | | 0110 _B | PCLK/2 ⁶ |
| | | 0111 _B | PCLK/2 ⁷ |
| | | 1000 _B | PCLK/2 ⁸ |
| | | 1001 _B | PCLK/2 ⁹ |
| | | 1010 _B | PCLK/2 ¹⁰ |
| | | 1011 _B | PCLK/2 ¹¹ |
| | | 1100 _B | PCLK/2 ¹² |
| | | 1101 _B | PCLK/2 ¹³ |
| | | 1110 _B | PCLK/2 ¹⁴ |
| 1111 _B | PCLK/2 ¹⁵ | | |
| The above bits are rewritable only when all the counters using CK2 are stopped (TAUDnTE.TAUDnTEm = 0). | | | |
| 7 to 4 | TAUDnPRS1 [3:0] | Specify the CK1 clock. | |
| | | TAUDnPRS1[3:0] | CK1 Clock |
| | | 0000 _B | PCLK/2 ⁰ |
| | | 0001 _B | PCLK/2 ¹ |
| | | 0010 _B | PCLK/2 ² |
| | | 0011 _B | PCLK/2 ³ |
| | | 0100 _B | PCLK/2 ⁴ |
| | | 0101 _B | PCLK/2 ⁵ |
| | | 0110 _B | PCLK/2 ⁶ |
| | | 0111 _B | PCLK/2 ⁷ |
| | | 1000 _B | PCLK/2 ⁸ |
| | | 1001 _B | PCLK/2 ⁹ |
| | | 1010 _B | PCLK/2 ¹⁰ |
| | | 1011 _B | PCLK/2 ¹¹ |
| | | 1100 _B | PCLK/2 ¹² |
| | | 1101 _B | PCLK/2 ¹³ |
| | | 1110 _B | PCLK/2 ¹⁴ |
| 1111 _B | PCLK/2 ¹⁵ | | |
| The above bits are rewritable only when all the counters using CK1 are stopped (TAUDnTE.TAUDnTEm = 0). | | | |

Table 23.11 TAUDnTPS Register Contents (3/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|----------------------|---|----------------|-----------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|
| 3 to 0 | TAUDnPRS0 [3:0] | Specify the CK0 clock. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUDnPRS0[3:0]</th> <th>CK0 Clock</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>PCLK/2⁰</td></tr> <tr><td>0001_B</td><td>PCLK/2¹</td></tr> <tr><td>0010_B</td><td>PCLK/2²</td></tr> <tr><td>0011_B</td><td>PCLK/2³</td></tr> <tr><td>0100_B</td><td>PCLK/2⁴</td></tr> <tr><td>0101_B</td><td>PCLK/2⁵</td></tr> <tr><td>0110_B</td><td>PCLK/2⁶</td></tr> <tr><td>0111_B</td><td>PCLK/2⁷</td></tr> <tr><td>1000_B</td><td>PCLK/2⁸</td></tr> <tr><td>1001_B</td><td>PCLK/2⁹</td></tr> <tr><td>1010_B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011_B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100_B</td><td>PCLK/2¹²</td></tr> <tr><td>1101_B</td><td>PCLK/2¹³</td></tr> <tr><td>1110_B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111_B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> | TAUDnPRS0[3:0] | CK0 Clock | 0000 _B | PCLK/2 ⁰ | 0001 _B | PCLK/2 ¹ | 0010 _B | PCLK/2 ² | 0011 _B | PCLK/2 ³ | 0100 _B | PCLK/2 ⁴ | 0101 _B | PCLK/2 ⁵ | 0110 _B | PCLK/2 ⁶ | 0111 _B | PCLK/2 ⁷ | 1000 _B | PCLK/2 ⁸ | 1001 _B | PCLK/2 ⁹ | 1010 _B | PCLK/2 ¹⁰ | 1011 _B | PCLK/2 ¹¹ | 1100 _B | PCLK/2 ¹² | 1101 _B | PCLK/2 ¹³ | 1110 _B | PCLK/2 ¹⁴ | 1111 _B | PCLK/2 ¹⁵ |
| TAUDnPRS0[3:0] | CK0 Clock | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | PCLK/2 ⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | PCLK/2 ¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | PCLK/2 ² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 _B | PCLK/2 ³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 _B | PCLK/2 ⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 _B | PCLK/2 ⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 _B | PCLK/2 ⁶ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 _B | PCLK/2 ⁷ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 _B | PCLK/2 ⁸ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 _B | PCLK/2 ⁹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 _B | PCLK/2 ¹⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 _B | PCLK/2 ¹¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 _B | PCLK/2 ¹² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 _B | PCLK/2 ¹³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 _B | PCLK/2 ¹⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | PCLK/2 ¹⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The above bits are rewritable only when all the counters using CK0 are stopped (TAUDnTE.TAUDnTEm = 0).

NOTE

The TAUDn clock input PCLK is specified in the first part of this section, **Section 23.1.3, Clock Supply**.

23.3.2.2 TAUDnBRS — TAUDn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUDnTPS.TAUDnPRS3[3:0].

Access: Readable/writable in 8-bit units. Write to this register when TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 244_H

Value after reset: 00_H This register is initialized by any reset source.

| | | | | | | | | |
|-------------------|---------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnBRS[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.12 TAUDnBRS Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 7 to 0 | TAUDnBRS[7:0] | Specify a CK3_PRE clock division factor for generating CK3. |
| | TAUDnBRS[7:0] | CK3 Clock |
| | 0000 0000 _B | CK3_PRE / 1 |
| | 0000 0001 _B | CK3_PRE / 2 |
| | 0000 0010 _B | CK3_PRE / 3 |
| | 0000 0011 _B | CK3_PRE / 4 |
| | ... | ... |
| | 1111 1110 _B | CK3_PRE / 255 |
| | 1111 1111 _B | CK3_PRE / 256 |

23.3.3 Details of TAUDn Control Registers

23.3.3.1 TAUDnCDRm — TAUDn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUDnCMORm.TAUDnMD[4:1].

Access: Readable/writable in 16-bit units.
When this register functions as a capture register, only reading is possible. Any write operation is ignored.
When this register functions as a compare register, reading and writing is possible.

Address: <TAUDn_base> + m × 4_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | TAUDnCDR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.13 TAUDnCDRm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|--|
| 15 to 0 | TAUDnCDR [15:0] | Data register for capture/compare values |

23.3.3.2 TAUDnCNTm — TAUDn Channel Counter Register

This is a channel m counter register.

Access: Readable in 16-bit units.

Address: <TAUDn_base> + 80_H + m × 4_H

Value after reset: FFFF_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | TAUDnCNT[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 23.14 TAUDnCNTm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|----------------------|
| 15 to 0 | TAUDnCNT [15:0] | 16-bit counter value |

A read value depends on a counter value, a changed operating mode, TAUDnTS.TAUDnTSM or TAUDnTT.TAUDnTTm bit value.

The initial read value of the counter depends on an operating mode and how the counter is stopped.

- Stop by a reset
- Stop by a counter stop trigger (TAUDnTT.TAUDnTTm = 1)

Table 23.15 lists the initial counter read values after the counter is stopped (TAUDnTE.TAUDnTEm = 0) and re-enabled (TAUDnTS.TAUDnTSM = 1).

The table also contains the counter read value one count after the counter is enabled (TAUDnTS.TAUDnTSM = 1) with the counter waiting for a start trigger.

Table 23.15 TAUDnCNTm Read Values after Re-Enabling Counter

| Mode Name | Count Method (Up/Down) | TAUDnCNTm Value | | |
|-----------------------------|------------------------|---------------------------|--------------------|-------------------------------|
| | | Start value* ¹ | After Stop Trigger | After One Count |
| Interval timer mode | Count down | FFFF _H | Stop value | — |
| Judge mode | Count down | FFFF _H | Stop value | — |
| Capture mode | Count up | 0000 _H | Stop value | — |
| Event count mode | Count down | FFFF _H | Stop value | — |
| One-count mode | Count down | FFFF _H | Stop value | Stop value |
| Capture and one-count mode | Count up | 0000 _H | Stop value | Capture value + 1 (TAUDnCDRm) |
| Judge and one-count mode | Count down | FFFF _H | Stop value | TAUDnCNTm value – 1 |
| Up/down count mode | Count down/up | FFFF _H | Stop value | — |
| Pulse one count mode | Count down | FFFF _H | Stop value | 0000 _H |
| Count capture mode | Count up | 0000 _H | Stop value | — |
| Capture and gate count mode | Count up | 0000 _H | Stop value | Stop value |

Note 1. The value set in TAUDnCNTm when operation mode is changes after reset release.

23.3.3.3 TAUDnCMORm — TAUDn Channel Mode OS Register

This register controls channel m operation.

Access: Readable/writable in 16-bit units. Write to this register when TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 200H + m × 4H

Value after reset: 0000H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------|-----|----------------|-----|----------|---------------|-----|----------------|-----|-----|--------------|-----|-----|-----|-----|-----|
| | TAUDnCKS [1:0] | | TAUDnCCS [1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS [1:0] | | — | TAUDnMD[4:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.16 TAUDnCMORm Register Contents (1/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|---------------|--|-----------|-----------|------------------------------|---|---|---|---|---|---------------------------------------|---|---|--------------------|---|---|-------------------------------------|
| 15, 14 | TAUDnCKS[1:0] | <p>Selects an operating clock. The operating clock is used for the following two operations.</p> <ul style="list-style-type: none"> Used for the TAUDnTTINm input edge detection circuit. Used as a counter clock as TAUDnCNTm by the TAUDnCMORm.TAUDnCCS[1:0] bit setting. <table border="1"> <thead> <tr> <th>TAUDnCKS1</th> <th>TAUDnCKS0</th> <th>Selection of Operating Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>CK0</td> </tr> <tr> <td>0</td> <td>1</td> <td>CK1</td> </tr> <tr> <td>1</td> <td>0</td> <td>CK2</td> </tr> <tr> <td>1</td> <td>1</td> <td>CK3</td> </tr> </tbody> </table> | TAUDnCKS1 | TAUDnCKS0 | Selection of Operating Clock | 0 | 0 | CK0 | 0 | 1 | CK1 | 1 | 0 | CK2 | 1 | 1 | CK3 |
| TAUDnCKS1 | TAUDnCKS0 | Selection of Operating Clock | | | | | | | | | | | | | | | |
| 0 | 0 | CK0 | | | | | | | | | | | | | | | |
| 0 | 1 | CK1 | | | | | | | | | | | | | | | |
| 1 | 0 | CK2 | | | | | | | | | | | | | | | |
| 1 | 1 | CK3 | | | | | | | | | | | | | | | |
| 13, 12 | TAUDnCCS[1:0] | <p>Selects a count clock for TAUDnCNTm counter.</p> <table border="1"> <thead> <tr> <th>TAUDnCCS1</th> <th>TAUDnCCS0</th> <th>Selection of Count Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Operating clock specified by TAUDnCMORm.TAUDnCKS[1:0]</td> </tr> <tr> <td>0</td> <td>1</td> <td>Valid edge of TAUDnTTINm input signal</td> </tr> <tr> <td>1</td> <td>0</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>1</td> <td>INTTAUDnIm signal of master channel</td> </tr> </tbody> </table> | TAUDnCCS1 | TAUDnCCS0 | Selection of Count Clock | 0 | 0 | Operating clock specified by TAUDnCMORm.TAUDnCKS[1:0] | 0 | 1 | Valid edge of TAUDnTTINm input signal | 1 | 0 | Setting prohibited | 1 | 1 | INTTAUDnIm signal of master channel |
| TAUDnCCS1 | TAUDnCCS0 | Selection of Count Clock | | | | | | | | | | | | | | | |
| 0 | 0 | Operating clock specified by TAUDnCMORm.TAUDnCKS[1:0] | | | | | | | | | | | | | | | |
| 0 | 1 | Valid edge of TAUDnTTINm input signal | | | | | | | | | | | | | | | |
| 1 | 0 | Setting prohibited | | | | | | | | | | | | | | | |
| 1 | 1 | INTTAUDnIm signal of master channel | | | | | | | | | | | | | | | |
| 11 | TAUDnMAS | <p>Specifies whether the channel is a master channel or slave channel during synchronous channel operation.</p> <p>0: Slave 1: Master</p> <p>This bit setting is valid only for even-numbered channels (CHm_even). Odd-numbered channels (CHm_odd) are fixed to 0.</p> | | | | | | | | | | | | | | | |

Table 23.16 TAUDnCMORm Register Contents (2/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---------------|--|---|-----------|-----------|------------------------|---|---|---|---|---|---|---|--|---|---|---|---|---|---|--|---|---|---|---|--|---|---|---|--|---|---|---|--|---|---|---|---|
| 10 to 8 | TAUDnSTS[2:0] | Selects an external start trigger. <table border="1" data-bbox="676 349 1422 813"> <thead> <tr> <th>TAUDnSTS2</th> <th>TAUDnSTS1</th> <th>TAUDnSTS0</th> <th>Functional description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Software trigger</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Valid edge of TAUDnTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0].</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Valid edge of TAUDnTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Triggers simultaneous rewrite.</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>INTTAUDnIm of master channel is used as a start trigger.</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>INTTAUDnIm of upper channel (m - 1) regardless of master setting is used as a start trigger.</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Dead time output signal of the TO controller at the TAUDnTTOUTm output</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Up/down output trigger signal of master channel</td> </tr> </tbody> </table> | TAUDnSTS2 | TAUDnSTS1 | TAUDnSTS0 | Functional description | 0 | 0 | 0 | Software trigger | 0 | 0 | 1 | Valid edge of TAUDnTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0]. | 0 | 1 | 0 | Valid edge of TAUDnTTINm input signal is used as a start trigger and the opposite edge as a stop trigger. | 0 | 1 | 1 | Triggers simultaneous rewrite. | 1 | 0 | 0 | INTTAUDnIm of master channel is used as a start trigger. | 1 | 0 | 1 | INTTAUDnIm of upper channel (m - 1) regardless of master setting is used as a start trigger. | 1 | 1 | 0 | Dead time output signal of the TO controller at the TAUDnTTOUTm output | 1 | 1 | 1 | Up/down output trigger signal of master channel |
| TAUDnSTS2 | TAUDnSTS1 | TAUDnSTS0 | Functional description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | Software trigger | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | Valid edge of TAUDnTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0]. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | Valid edge of TAUDnTTINm input signal is used as a start trigger and the opposite edge as a stop trigger. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | Triggers simultaneous rewrite. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | INTTAUDnIm of master channel is used as a start trigger. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | INTTAUDnIm of upper channel (m - 1) regardless of master setting is used as a start trigger. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | Dead time output signal of the TO controller at the TAUDnTTOUTm output | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | Up/down output trigger signal of master channel | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7, 6 | TAUDnCOS[1:0] | Specify the timing for updating capture register TAUDnCDRm and overflow flag TAUDnCSRm.TAUDnOVF of channel m. These bits are valid only when channel m is in capture modes (capture and capture & one-count). <ul style="list-style-type: none"> • Capture mode • Capture and one-count mode • Capture and gate count mode • Count capture mode <table border="1" data-bbox="676 1093 1422 1861"> <thead> <tr> <th>TAUDnCOS1</th> <th>TAUDnCOS0</th> <th>TAUDnCDRm</th> <th>TAUDnCSRm.TAUDnOVF</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Updated upon detection of valid edge of TAUDnTTINm input.</td> <td>Updated (cleared or set) by detecting valid edge of TAUDnTTINm input: <ul style="list-style-type: none"> • If a counter overflow has occurred since the last detection of valid edge, set TAUDnCSRm.TAUDnOVF. • If no counter overflow has occurred since the last detection of valid edge, clear TAUDnCSRm.TAUDnOVF. </td> </tr> <tr> <td>0</td> <td>1</td> <td></td> <td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Updated upon detection of valid edge of TAUDnTTINm input and at the occurrence of counter overflow:</td> <td>Not operating</td> </tr> <tr> <td>1</td> <td>1</td> <td> Updated upon detection of valid edge of TAUDnTTINm input and at the occurrence of counter overflow: <ul style="list-style-type: none"> • Detection of valid edge of TAUDnTTINm input: Counter value is written into TAUDnCDRm. • Occurrence of overflow: FFFF_H is loaded into TAUDnCDRm. The next detection of valid edge of TAUDnTTINm input is ignored. </td> <td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td> </tr> </tbody> </table> | TAUDnCOS1 | TAUDnCOS0 | TAUDnCDRm | TAUDnCSRm.TAUDnOVF | 0 | 0 | Updated upon detection of valid edge of TAUDnTTINm input. | Updated (cleared or set) by detecting valid edge of TAUDnTTINm input: <ul style="list-style-type: none"> • If a counter overflow has occurred since the last detection of valid edge, set TAUDnCSRm.TAUDnOVF. • If no counter overflow has occurred since the last detection of valid edge, clear TAUDnCSRm.TAUDnOVF. | 0 | 1 | | Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1. | 1 | 0 | Updated upon detection of valid edge of TAUDnTTINm input and at the occurrence of counter overflow: | Not operating | 1 | 1 | Updated upon detection of valid edge of TAUDnTTINm input and at the occurrence of counter overflow: <ul style="list-style-type: none"> • Detection of valid edge of TAUDnTTINm input: Counter value is written into TAUDnCDRm. • Occurrence of overflow: FFFF_H is loaded into TAUDnCDRm. The next detection of valid edge of TAUDnTTINm input is ignored. | Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1. | | | | | | | | | | | | | | | | |
| TAUDnCOS1 | TAUDnCOS0 | TAUDnCDRm | TAUDnCSRm.TAUDnOVF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Updated upon detection of valid edge of TAUDnTTINm input. | Updated (cleared or set) by detecting valid edge of TAUDnTTINm input: <ul style="list-style-type: none"> • If a counter overflow has occurred since the last detection of valid edge, set TAUDnCSRm.TAUDnOVF. • If no counter overflow has occurred since the last detection of valid edge, clear TAUDnCSRm.TAUDnOVF. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | | Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Updated upon detection of valid edge of TAUDnTTINm input and at the occurrence of counter overflow: | Not operating | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Updated upon detection of valid edge of TAUDnTTINm input and at the occurrence of counter overflow: <ul style="list-style-type: none"> • Detection of valid edge of TAUDnTTINm input: Counter value is written into TAUDnCDRm. • Occurrence of overflow: FFFF_H is loaded into TAUDnCDRm. The next detection of valid edge of TAUDnTTINm input is ignored. | Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 23.16 TAUDnCMORm Register Contents (3/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|---|-----------|----------------------|---|---|--|--|--|--|---|---|--|---|---|---|---|---|-----|------------|---|---|---|---|-----|--------------|---|---|---|---|---|------------------|---|---|---|---|-----|----------------|---|---|---|---|-----|--------------------|---|---|---|---|---|----------------------------|---|---|---|---|-----|--------------------------|---|---|---|---|---|--------------------|---|---|---|---|---|--------------------|---|---|---|---|-----|----------------------|---|---|---|---|-----|--------------------|---|---|---|---|---|--------------------|---|---|---|---|---|-----------------------------|------------------|--|--|--|--|--------------------|
| 4 to 0 | TAUDnMD[4:0] | Specify an operating mode. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUDn MD4</th> <th>TAUDn MD3</th> <th>TAUDn MD2</th> <th>TAUDn MD1</th> <th>TAUDn MD0</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Judge mode</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Event count mode</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1/0</td><td>Judge and one-count mode</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Up/down count mode</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Pulse one-count mode</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr> <tr><td colspan="5" style="text-align: center;">Other than above</td><td>Setting prohibited</td></tr> </tbody> </table> | TAUDn MD4 | TAUDn MD3 | TAUDn MD2 | TAUDn MD1 | TAUDn MD0 | Functional Description | 0 | 0 | 0 | 0 | 1/0 | Interval timer mode | 0 | 0 | 0 | 1 | 1/0 | Judge mode | 0 | 0 | 1 | 0 | 1/0 | Capture mode | 0 | 0 | 1 | 1 | 0 | Event count mode | 0 | 1 | 0 | 0 | 1/0 | One-count mode | 0 | 1 | 0 | 1 | 1/0 | Setting prohibited | 0 | 1 | 1 | 0 | 0 | Capture and one-count mode | 0 | 1 | 1 | 1 | 1/0 | Judge and one-count mode | 1 | 0 | 0 | 0 | 0 | Setting prohibited | 1 | 0 | 0 | 1 | 0 | Up/down count mode | 1 | 0 | 1 | 0 | 1/0 | Pulse one-count mode | 1 | 0 | 1 | 1 | 1/0 | Count capture mode | 1 | 1 | 0 | 0 | 0 | Setting prohibited | 1 | 1 | 0 | 1 | 0 | Capture and gate count mode | Other than above | | | | | Setting prohibited |
| TAUDn MD4 | TAUDn MD3 | TAUDn MD2 | TAUDn MD1 | TAUDn MD0 | Functional Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 1/0 | Interval timer mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1/0 | Judge mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 1/0 | Capture mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 0 | Event count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 1/0 | One-count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 1/0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 0 | Capture and one-count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 1/0 | Judge and one-count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 1 | 0 | Up/down count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | 1/0 | Pulse one-count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | 1/0 | Count capture mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 1 | 0 | Capture and gate count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Other than above | | | | | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>Mode</th> <th>Role of TAUDnMD0 Bit</th> </tr> </thead> <tbody> <tr> <td>Interval timer mode Capture mode Count capture mode</td> <td>Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated.</td> </tr> <tr> <td>Event count mode Up/down count mode</td> <td>This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation).</td> </tr> <tr> <td>One-count mode Pulse one-count mode</td> <td>Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. CAUTION • In one-count mode, INTTAUDnIm signal is not output at the beginning of count operation. • In pulse one-count mode, INTTAUDnIm signal is output at the beginning of count operation.</td> </tr> <tr> <td>Capture and one-count mode Capture and gate count mode</td> <td>This bit should be set to 0. CAUTION INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.</td> </tr> <tr> <td>Judge mode Judge and one-count mode</td> <td>Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm > TAUDnCDRm</td> </tr> </tbody> </table> | Mode | Role of TAUDnMD0 Bit | Interval timer mode Capture mode Count capture mode | Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated. | Event count mode Up/down count mode | This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation). | One-count mode Pulse one-count mode | Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. CAUTION • In one-count mode, INTTAUDnIm signal is not output at the beginning of count operation. • In pulse one-count mode, INTTAUDnIm signal is output at the beginning of count operation. | Capture and one-count mode Capture and gate count mode | This bit should be set to 0. CAUTION INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled. | Judge mode Judge and one-count mode | Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm > TAUDnCDRm | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mode | Role of TAUDnMD0 Bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Interval timer mode Capture mode Count capture mode | Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Event count mode Up/down count mode | This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation). | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| One-count mode Pulse one-count mode | Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. CAUTION • In one-count mode, INTTAUDnIm signal is not output at the beginning of count operation. • In pulse one-count mode, INTTAUDnIm signal is output at the beginning of count operation. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Capture and one-count mode Capture and gate count mode | This bit should be set to 0. CAUTION INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Judge mode Judge and one-count mode | Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm > TAUDnCDRm | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

23.3.3.4 TAUDnCMURm — TAUDn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUDnTTINm input.

Access: Readable/writable in 8-bit units.

Address: <TAUDn_base> + C0_H + m × 4_H

Value after reset: 00_H This register is initialized by any reset source.

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 23.17 TAUDnCMURm Register Contents

| Bit position | Bit name | Function | | | | | | | | | | | | | | | |
|---|---------------|--|-----------|-----------|------------------------|---|---|--------------|---|---|-------------|---|---|--|---|---|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 1, 0 | TAUDnTIS[1:0] | Specify a valid edge of TAUDnTTINm input signal. | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUDnTIS1</th> <th>TAUDnTIS0</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Falling edge</td> </tr> <tr> <td>0</td> <td>1</td> <td>Rising edge</td> </tr> <tr> <td>1</td> <td>0</td> <td>Detection of rising and falling edges (selects low width measurement) <ul style="list-style-type: none"> Start trigger: Falling edge Stop trigger (capture): Rising edge </td> </tr> <tr> <td>1</td> <td>1</td> <td>Detection of rising and falling edges (selects high width measurement) <ul style="list-style-type: none"> Start trigger: Rising edge Stop trigger (capture): Falling edge </td> </tr> </tbody> </table> | TAUDnTIS1 | TAUDnTIS0 | Functional Description | 0 | 0 | Falling edge | 0 | 1 | Rising edge | 1 | 0 | Detection of rising and falling edges (selects low width measurement) <ul style="list-style-type: none"> Start trigger: Falling edge Stop trigger (capture): Rising edge | 1 | 1 | Detection of rising and falling edges (selects high width measurement) <ul style="list-style-type: none"> Start trigger: Rising edge Stop trigger (capture): Falling edge |
| TAUDnTIS1 | TAUDnTIS0 | Functional Description | | | | | | | | | | | | | | | |
| 0 | 0 | Falling edge | | | | | | | | | | | | | | | |
| 0 | 1 | Rising edge | | | | | | | | | | | | | | | |
| 1 | 0 | Detection of rising and falling edges (selects low width measurement) <ul style="list-style-type: none"> Start trigger: Falling edge Stop trigger (capture): Rising edge | | | | | | | | | | | | | | | |
| 1 | 1 | Detection of rising and falling edges (selects high width measurement) <ul style="list-style-type: none"> Start trigger: Rising edge Stop trigger (capture): Falling edge | | | | | | | | | | | | | | | |
| Edge detection of TAUDnTTINm input signal is based on the operating clock selected by TAUDnCMORm.TAUDnCKS[1:0]. | | | | | | | | | | | | | | | | | |

23.3.3.5 TAUDnCSRm — TAUDn channel status register

This register indicates the up/down count and overflow status of channel m counter.

Access: Readable in 8-bit units.

Address: <TAUDn_base> + 140H + m × 4H

Value after reset: 00H This register is initialized by any reset source.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|----------|----------|
| | — | — | — | — | — | — | TAUDnCSF | TAUDnOVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 23.18 TAUDnCSRm Register Contents

| Bit position | Bit name | Function |
|--------------|----------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. |
| 1 | TAUDnCSF | Indicates a count direction. 0: Count-up 1: Count-down The read value of this bit is valid only in up/down count mode. In other modes, an undefined value is read. |
| 0 | TAUDnOVF | Indicates counter overflow status. 0: No overflow occurs. 1: Overflow occurs. This bit is used only in the following modes: <ul style="list-style-type: none"> • Capture mode • Capture and one-count mode <p>The function of this bit depends on the setting of control bit TAUDnCMORm.TAUDnCOS[1:0]. In other modes, an undefined value is read.</p> |

23.3.3.6 TAUDnCSm - TAUDn Channel Status Clear Register

This is a trigger register for clearing the overflow flag TAUDnCSRm.TAUDnOVF of channel m.

Access: Writable in 8-bit units. This value is always read as 00_H.

Address: <TAUDn_base> + 180_H + m × 4_H

Value after reset: 00_H This register is initialized by any reset source.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------|
| | — | — | — | — | — | — | — | TAUDnCLOV |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 23.19 TAUDnCSm Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TAUDnCLOV | This bit clears the overflow flag of channel m. Writing 1 to this bit clears the overflow flag TAUDnCSRm.TAUDnOVF, though writing 0 is ignored. |

23.3.3.7 TAUDnTS — TAUDn Channel Start Trigger Register

This register enables the counter operation of each channel.

Access: Writable in 16-bit units. This value is always read as 0000_H.

Address: <TAUDn_base> + 1C4_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | TAUDnTS15 | TAUDnTS14 | TAUDnTS13 | TAUDnTS12 | TAUDnTS11 | TAUDnTS10 | TAUDnTS09 | TAUDnTS08 | TAUDnTS07 | TAUDnTS06 | TAUDnTS05 | TAUDnTS04 | TAUDnTS03 | TAUDnTS02 | TAUDnTS01 | TAUDnTS00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 23.20 TAUDnTS Register Contents

| Bit position | Bit name | Function |
|--------------|----------|--|
| 15 to 0 | TAUDnTSM | Enables the counter operation of channel m. Writing 1 to this bit enables the counter operation and sets TAUDnTE.TAUDnTEm to 1, though writing 0 is ignored. |

23.3.3.8 TAUDnTE — TAUDn Channel Enable Status Register

This register indicates the enable/disable state of the counter operation.

Access: Readable in 16-bit units.

Address: <TAUDn_base> + 1C0_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | TAUDnTE15 | TAUDnTE14 | TAUDnTE13 | TAUDnTE12 | TAUDnTE11 | TAUDnTE10 | TAUDnTE09 | TAUDnTE08 | TAUDnTE07 | TAUDnTE06 | TAUDnTE05 | TAUDnTE04 | TAUDnTE03 | TAUDnTE02 | TAUDnTE01 | TAUDnTE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 23.21 TAUDnTE Register Contents

| Bit position | Bit name | Function |
|--------------|----------------------|--|
| 15 to 0 | TAUDnTE _m | Indicates the enable/disable state of the counter operation of channel m. 0: The counter operation is disabled. 1: The counter operation is enabled. This bit is set to 1 when trigger input of TAUDnTSST _m (synchronous channel start trigger signal) is detected or when TAUDnTS.TAUDnTS _m is set to 1. This bit is set to 0 when TAUDnTT.TAUDnTT _m is set to 1. |

23.3.3.9 TAUDnTT — TAUDn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: Writable in 16-bit units. This value is always read as 0000_H.

Address: <TAUDn_base> + 1C8_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | TAUDnTT15 | TAUDnTT14 | TAUDnTT13 | TAUDnTT12 | TAUDnTT11 | TAUDnTT10 | TAUDnTT09 | TAUDnTT08 | TAUDnTT07 | TAUDnTT06 | TAUDnTT05 | TAUDnTT04 | TAUDnTT03 | TAUDnTT02 | TAUDnTT01 | TAUDnTT00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 23.22 TAUDnTT Register Contents

| Bit position | Bit name | Function |
|--------------|----------------------|---|
| 15 to 0 | TAUDnTT _m | Stops the counter operation of channel m. Writing 1 to this bit stops the counter operation and clears TAUDnTE.TAUDnTE _m to 0, though writing 0 is ignored. TAUDnCNT _m , TAUDnTO.TAUDnTO _m , and TAUDnTTO _m retain the values provided before the counter is stopped. |

23.3.4 Details of TAUDn Simultaneous Rewrite Registers

23.3.4.1 TAUDnRDE — TAUDn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of TAUDnCDRm/TAUDnTOLm data register.

Access: Readable/writable in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 260_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnRDE15 | TAUDnRDE14 | TAUDnRDE13 | TAUDnRDE12 | TAUDnRDE11 | TAUDnRDE10 | TAUDnRDE09 | TAUDnRDE08 | TAUDnRDE07 | TAUDnRDE06 | TAUDnRDE05 | TAUDnRDE04 | TAUDnRDE03 | TAUDnRDE02 | TAUDnRDE01 | TAUDnRDE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.23 TAUDnRDE Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|--|
| 15 to 0 | TAUDnRDEm | Enables/disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite (Loaded by the detection of its own channel match) 1: Enables simultaneous rewrite |

23.3.4.2 TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register

This register selects a channel that controls simultaneous rewrite.

Access: Readable/writable in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 268_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnRDS15 | TAUDnRDS14 | TAUDnRDS13 | TAUDnRDS12 | TAUDnRDS11 | TAUDnRDS10 | TAUDnRDS09 | TAUDnRDS08 | TAUDnRDS07 | TAUDnRDS06 | TAUDnRDS05 | TAUDnRDS04 | TAUDnRDS03 | TAUDnRDS02 | TAUDnRDS01 | TAUDnRDS00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.24 TAUDnRDS Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|--|
| 15 to 0 | TAUDnRDSm | Selects a channel that controls a simultaneous rewrite trigger. 0: Master channel 1: Another upper channel (The channel when TAUDnRDE.TAUDnRDEm = 1) |

23.3.4.3 TAUDnRDM — TAUDn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

Access: Readable/writable in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 264_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnRDM15 | TAUDnRDM14 | TAUDnRDM13 | TAUDnRDM12 | TAUDnRDM11 | TAUDnRDM10 | TAUDnRDM09 | TAUDnRDM08 | TAUDnRDM07 | TAUDnRDM06 | TAUDnRDM05 | TAUDnRDM04 | TAUDnRDM03 | TAUDnRDM02 | TAUDnRDM01 | TAUDnRDM00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.25 TAUDnRDM Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnRDMm | Selects the timing for generating a simultaneous rewrite trigger signal. 0: When the master channel counter starts to count 1: At the peak of cycle of triangular wave These bit settings are applied to the channel when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 0. |

23.3.4.4 TAUDnRDC — TAUDn Channel Reload Data Control Register

This register specifies a channel which generates an INTTAUDnIm signal to trigger simultaneous rewrite.

Access: Readable/writable in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 26C_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnRDC15 | TAUDnRDC14 | TAUDnRDC13 | TAUDnRDC12 | TAUDnRDC11 | TAUDnRDC10 | TAUDnRDC09 | TAUDnRDC08 | TAUDnRDC07 | TAUDnRDC06 | TAUDnRDC05 | TAUDnRDC04 | TAUDnRDC03 | TAUDnRDC02 | TAUDnRDC01 | TAUDnRDC00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.26 TAUDnRDC Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnRDCm | Specify whether the channel generates a simultaneous rewrite trigger signal or not. 0: Not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bit settings are applied to the channel when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 1. |

23.3.4.5 TAUDnRDT — TAUDn Channel Reload Data Trigger Register

This register triggers the enabling state of simultaneous rewrite.

Access: Writable in 16-bit units. This value is always read as 0000_H.

Address: <TAUDn_base> + 044_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnRDT15 | TAUDnRDT14 | TAUDnRDT13 | TAUDnRDT12 | TAUDnRDT11 | TAUDnRDT10 | TAUDnRDT09 | TAUDnRDT08 | TAUDnRDT07 | TAUDnRDT06 | TAUDnRDT05 | TAUDnRDT04 | TAUDnRDT03 | TAUDnRDT02 | TAUDnRDT01 | TAUDnRDT00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 23.27 TAUDnRDT Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnRDTm | These bits specify the trigger to the enabling state of simultaneous rewrite. 0: No function 1: The simultaneous rewrite enabling flag (TAUDnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. The setting of these bits is applied to the channel when TAUDnRDE.TAUDnRDEm is 1. |

23.3.4.6 TAUDnRSF — TAUDn Channel Reload Status Register

This flag register indicates simultaneous rewrite status.

Access: Readable in 16-bit units.

Address: <TAUDn_base> + 048_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnRSF15 | TAUDnRSF14 | TAUDnRSF13 | TAUDnRSF12 | TAUDnRSF11 | TAUDnRSF10 | TAUDnRSF09 | TAUDnRSF08 | TAUDnRSF07 | TAUDnRSF06 | TAUDnRSF05 | TAUDnRSF04 | TAUDnRSF03 | TAUDnRSF02 | TAUDnRSF01 | TAUDnRSF00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 23.28 TAUDnRSF Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnRSFm | Indicates simultaneous rewrite status. 0: Indicates that a simultaneous rewrite has completed by the generation of a simultaneous rewrite trigger. 1: Indicates that a simultaneous rewrite trigger is waited in a simultaneous rewrite enable state (TAUDnRDTm = 1). |

23.3.5 Details of TAUDn Output Registers

23.3.5.1 TAUDnTOE — TAUDn Channel Output Enable Register

This register enables/disables the independent channel output mode controlled by software.

Access: Readable/writable in 16-bit units.

Address: <TAUDn_base> + 5C_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTOE15 | TAUDnTOE14 | TAUDnTOE13 | TAUDnTOE12 | TAUDnTOE11 | TAUDnTOE10 | TAUDnTOE09 | TAUDnTOE08 | TAUDnTOE07 | TAUDnTOE06 | TAUDnTOE05 | TAUDnTOE04 | TAUDnTOE03 | TAUDnTOE02 | TAUDnTOE01 | TAUDnTOE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.29 TAUDnTOE Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnTOEm | Enables or disables the independent channel output mode. 0: Disables the independent timer output function (software control) 1: Enables the independent timer output function. |

23.3.5.2 TAUDnTO — TAUDn Channel Output Register

This register specifies and reads a TAUDnTTOUTm level.

Access: Readable/writable in 16-bit units.

Address: <TAUDn_base> + 58_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTOO15 | TAUDnTOO14 | TAUDnTOO13 | TAUDnTOO12 | TAUDnTOO11 | TAUDnTOO10 | TAUDnTOO09 | TAUDnTOO08 | TAUDnTOO07 | TAUDnTOO06 | TAUDnTOO05 | TAUDnTOO04 | TAUDnTOO03 | TAUDnTOO02 | TAUDnTOO01 | TAUDnTOO00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.30 TAUDnTO Register Contents

| Bit position | Bit name | Function |
|--------------|----------|--|
| 15 to 0 | TAUDnTOM | Specify and reads a TAUDnTTOUTm level. 0: Low level 1: High level Writing to these bits is enabled only when TAUDnTOE.TAUDnTOEm is 0. |

23.3.5.3 TAUDnTOM — TAUDn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTE_m = 0).

Address: <TAUDn_base> + 248_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTOM15 | TAUDnTOM14 | TAUDnTOM13 | TAUDnTOM12 | TAUDnTOM11 | TAUDnTOM10 | TAUDnTOM09 | TAUDnTOM08 | TAUDnTOM07 | TAUDnTOM06 | TAUDnTOM05 | TAUDnTOM04 | TAUDnTOM03 | TAUDnTOM02 | TAUDnTOM01 | TAUDnTOM00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.31 TAUDnTOM Register Contents

| Bit position | Bit name | Function |
|--------------|-----------------------|---|
| 15 to 0 | TAUDnTOM _m | Specify an output mode. 0: Independent channel operation 1: Synchronous channel operation |

23.3.5.4 TAUDnTOC — TAUDn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUDnTOM.TAUDnTOM_m.

Access: Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTE_m = 0).

Address: <TAUDn_base> + 24C_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTOC15 | TAUDnTOC14 | TAUDnTOC13 | TAUDnTOC12 | TAUDnTOC11 | TAUDnTOC10 | TAUDnTOC09 | TAUDnTOC08 | TAUDnTOC07 | TAUDnTOC06 | TAUDnTOC05 | TAUDnTOC04 | TAUDnTOC03 | TAUDnTOC02 | TAUDnTOC01 | TAUDnTOC00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.32 TAUDnTOC Register Contents

| Bit position | Bit name | Function | |
|--------------|------------------------|--|---|
| 15 to 0 | TAUDnTOC _m | Specify an output mode. 0: Operating mode 1 1: Operating mode 2 These bits specify the operating mode of TAUDnTOUT _m by the setting of TAUDnTOM.TAUDnTOM _m and TAUDnTOC.TAUDnTOC _m . | |
| | TOM_m | TOC_m | Functional Description |
| | 0 | 0 | Toggle mode: Toggle operation is conducted when INTTAUDnIm occurs. |
| | 0 | 1 | Set/reset mode: Set when INTTAUDnIm occurs at the beginning of count operation, and reset when INTTAUDnIm is caused by detection of a match between TAUDnCNT _m and TAUDnCDR _m . |
| | 1 | 0 | Synchronous channel operating mode 1: Set when INT occurs on master channels, and reset when INT occurs on slave channels. |
| | 1 | 1 | Synchronous channel operating mode 2: <ul style="list-style-type: none"> Set when INTTAUDnIm occurs in count-down status. Reset when INTTAUDnIm occurs in count-up status. |

23.3.5.5 TAUDnTOL — TAUDn Channel Output Active Level Register

This register specifies the output logic of channel output bit (TAUDnTO.TAUDnTOm).

Access: Readable/writable in 16-bit units.

Address: <TAUDn_base> + 040H

Value after reset: 0000H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTOL15 | TAUDnTOL14 | TAUDnTOL13 | TAUDnTOL12 | TAUDnTOL11 | TAUDnTOL10 | TAUDnTOL09 | TAUDnTOL08 | TAUDnTOL07 | TAUDnTOL06 | TAUDnTOL05 | TAUDnTOL04 | TAUDnTOL03 | TAUDnTOL02 | TAUDnTOL01 | TAUDnTOL00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.33 TAUDnTOL Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnTOLm | Specify the output logic of channel m output bit (TAUDnTO.TAUDnTOm). 0: Positive logic (active high) 1: Negative logic (active low) The value of these bits are enabled when the combination of following three conditions is enabled and TAUDnTOE.TAUDnTOEm = 1. <ul style="list-style-type: none"> • TAUDnTOM.TAUDnTOMm = 0, TAUDnTOC.TAUDnTOCm = 1 • TAUDnTOM.TAUDnTOMm = 1, TAUDnTOC.TAUDnTOCm = 1 • TAUDnTOM.TAUDnTOMm = 1, TAUDnTOC.TAUDnTOCm = 0 |

23.3.6 Details of TAUDn Dead Time Output Registers

23.3.6.1 TAUDnTDE — TAUDn Channel Dead Time Output Enable Register

This register enables/disables the dead time operation of every channel.

Access: Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTE_m = 0).

Address: <TAUDn_base> + 250_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTDE15 | TAUDnTDE14 | TAUDnTDE13 | TAUDnTDE12 | TAUDnTDE11 | TAUDnTDE10 | TAUDnTDE09 | TAUDnTDE08 | TAUDnTDE07 | TAUDnTDE06 | TAUDnTDE05 | TAUDnTDE04 | TAUDnTDE03 | TAUDnTDE02 | TAUDnTDE01 | TAUDnTDE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.34 TAUDnTDE Register Contents

| Bit position | Bit name | Function |
|--------------|-----------------------|--|
| 15 to 0 | TAUDnTDE _m | Enables/disables the dead time control operation of channel m. 0: Disables dead time operation 1: Enables dead time operation. The same setting should be made for both even- and odd-numbered slave channels in pairs. These bit settings are applied when: <ul style="list-style-type: none"> TAUDnTOE.TAUDnTOE_m = 1, TAUDnTOM.TAUDnTOM_m = 1, TAUDnTOC.TAUDnTOC_m = 1 |

23.3.6.2 TAUDnTDM — TAUDn Channel Dead Time Output Mode Register

This register specifies the timing to add dead time during dead time output.

Access: Readable/writable in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTE_m = 0).

Address: <TAUDn_base> + 254_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTDM15 | TAUDnTDM14 | TAUDnTDM13 | TAUDnTDM12 | TAUDnTDM11 | TAUDnTDM10 | TAUDnTDM09 | TAUDnTDM08 | TAUDnTDM07 | TAUDnTDM06 | TAUDnTDM05 | TAUDnTDM04 | TAUDnTDM03 | TAUDnTDM02 | TAUDnTDM01 | TAUDnTDM00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.35 TAUDnTDM Register Contents

| Bit position | Bit name | Function |
|--------------|-----------------------|---|
| 15 to 0 | TAUDnTDM _m | Specify the timing to add dead time during dead time output. 0: When detecting the duty cycle of an upper even-numbered channel (duty dead time output). 1: When detecting the TAUDnTTIN _m input edge of a lower odd-numbered channel (one-phase dead time output). The same setting should be made for both even- and odd-numbered slave channels in pairs. These bit settings are applied when: <ul style="list-style-type: none"> TAUDnTOE.TAUDnTOE_m = 1, TAUDnTOM.TAUDnTOM_m = 1, TAUDnTOC.TAUDnTOC_m = 1, TAUDnTDE.TAUDnTDE_m = 1 |

23.3.6.3 TAUDnTDL — TAUDn Channel Dead Time Output Level Register

This register selects a phase in which dead time is added.

Access: Readable/writable in 16-bit units.

Address: <TAUDn_base> + 54_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTDL15 | TAUDnTDL14 | TAUDnTDL13 | TAUDnTDL12 | TAUDnTDL11 | TAUDnTDL10 | TAUDnTDL09 | TAUDnTDL08 | TAUDnTDL07 | TAUDnTDL06 | TAUDnTDL05 | TAUDnTDL04 | TAUDnTDL03 | TAUDnTDL02 | TAUDnTDL01 | TAUDnTDL00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.36 TAUDnTDL Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|--|
| 15 to 0 | TAUDnTDLm | Selects a phase in which dead time is added. 0: Normal phase 1: Reverse phase These bit settings are applied when: <ul style="list-style-type: none"> TAUDnTOE.TAUDnTOEm = 1, TAUDnTOM.TAUDnTOMm = 1, TAUDnTOC.TAUDnTOCm = 1, TAUDnTDE.TAUDnTDEm = 1 |

23.3.7 Details of TAUDn Real-time/Modulation Output Registers

23.3.7.1 TAUDnTRE — TAUDn Channel Real-time Output Enable Register

This register enables/disables real-time output.

Access: Readable/writable in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 258_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTRE15 | TAUDnTRE14 | TAUDnTRE13 | TAUDnTRE12 | TAUDnTRE11 | TAUDnTRE10 | TAUDnTRE09 | TAUDnTRE08 | TAUDnTRE07 | TAUDnTRE06 | TAUDnTRE05 | TAUDnTRE04 | TAUDnTRE03 | TAUDnTRE02 | TAUDnTRE01 | TAUDnTRE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.37 TAUDnTRE Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnTREm | Enables or disables real-time output of channel m. 0: Disables real-time output 1: Enables real-time output. These bit settings are applied only when TAUDnTOE.TAUDnTOEm = 1. When TAUDnTRE.TAUDnTREm = 0, TAUDnTTOUTm is not affected by real-time output. When TAUDnTRE.TAUDnTREm = 1, TAUDnTTOUTm outputs the value of real-time output bit TAUDnTRO.TAUDnTROm in response to a timer operation. |

23.3.7.2 TAUDnTRC — TAUDn Channel Real-time Output Control Register

This register controls the real-time output trigger of each channel.

Access: Readable/writable in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 25C_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTRC15 | TAUDnTRC14 | TAUDnTRC13 | TAUDnTRC12 | TAUDnTRC11 | TAUDnTRC10 | TAUDnTRC09 | TAUDnTRC08 | TAUDnTRC07 | TAUDnTRC06 | TAUDnTRC05 | TAUDnTRC04 | TAUDnTRC03 | TAUDnTRC02 | TAUDnTRC01 | TAUDnTRC00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.38 TAUDnTRC Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnTRCm | Specify a channel on which the real-time output trigger for channel m is generated. 0: Next upper channel with this bit set to 1 1: Channel m These bit settings are applied only when TAUDnTRE.TAUDnTREm = 1. |

23.3.7.3 TAUDnTRO — TAUDn Channel Real-time Output Register

This register sets a value which is output to TAUDnTTOUTm.

Access: Readable/writable in 16-bit units.

Address: <TAUDn_base> + 04C_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTRO15 | TAUDnTRO14 | TAUDnTRO13 | TAUDnTRO12 | TAUDnTRO11 | TAUDnTRO10 | TAUDnTRO09 | TAUDnTRO08 | TAUDnTRO07 | TAUDnTRO06 | TAUDnTRO05 | TAUDnTRO04 | TAUDnTRO03 | TAUDnTRO02 | TAUDnTRO01 | TAUDnTRO00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.39 TAUDnTRO Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|--|
| 15 to 0 | TAUDnTROm | Sets a value which is output to TAUDnTTOUTm. 0: Low 1: High TAUDnTROm value is not output to TAUDnTTOUTm when TAUDnTRE.TAUDnTREm = 0, even if a real-time output trigger occurs. |

23.3.7.4 TAUDnTME — TAUDn Channel Modulation Output Enable Register

This register enables/disables modulation output for timer output and real-time output.

Access: Readable/writable in 16-bit units.

Address: <TAUDn_base> + 050_H

Value after reset: 0000_H This register is initialized by any reset source.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | TAUDnTME15 | TAUDnTME14 | TAUDnTME13 | TAUDnTME12 | TAUDnTME11 | TAUDnTME10 | TAUDnTME09 | TAUDnTME08 | TAUDnTME07 | TAUDnTME06 | TAUDnTME05 | TAUDnTME04 | TAUDnTME03 | TAUDnTME02 | TAUDnTME01 | TAUDnTME00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 23.40 TAUDnTME Register Contents

| Bit position | Bit name | Function |
|--------------|-----------|---|
| 15 to 0 | TAUDnTMEm | Enables/disables modulation output for timer output and real-time output of channel m. 0: Disables modulation 1: Enables modulation These bit settings are applied only when TAUDnTOE.TAUDnTOEm and TAUDnTRE.TAUDnTREm = 1. |

23.4 Operating Procedure

The following lists the general operation procedure for the TAUDn:

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUDnTTOUTm is also initialized and outputs a low level.

- (1) Set the TAUDnTPS and TAUDnBRS registers to specify the clock frequency of CK0 to CK3.
- (2) Configure the desired TAUDn function:
 - Set the operation mode
 - Set the channel output mode
 - Set any other control bits
- (3) Enable the counter by setting the TAUDnTS.TAUDnTSM bit to 1.
The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
- (4) If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUDnTT.TAUDnTTm bit to 1. The counter can be forcibly restarted by setting the TAUDnTS.TAUDnTSM bit to 1.
- (5) Stop the function by setting the TAUDnTT.TAUDnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions is given below:
 - **Section 23.12, Independent Channel Operation Functions**
 - **Section 23.13, Synchronous Channel Operation Functions**
2. Change function while count operation is stopped (TAUDnTE.TAUDnTEm = 0).

23.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels). Several rules apply to the settings of channels. These rules are detailed in **Section 23.5.1, Rules of Synchronous Channel Operation**.

Two special features for synchronous channel operation are detailed in the following subchapters:

- **Section 23.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 23.6, Simultaneous Rewrite**

23.5.1 Rules of Synchronous Channel Operation

Number of master and slave channels

- Only even-numbered channels (CH0, CH2, CH4, ...) can be set as master channels. Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
Example: If CH2 is a master channel, CH3 and the lower channels (CH4, CH5, ...) can be set as slave channels.
- If multiple master channels are used, slave channels cannot cross the master channels.
Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operating clock

- The same operating clock must be set for the slave channel and the master channel. This is achieved by setting the TAUDnCMORm.TAUDnCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave usage and operating clocks are illustrated in **Figure 23.3**.

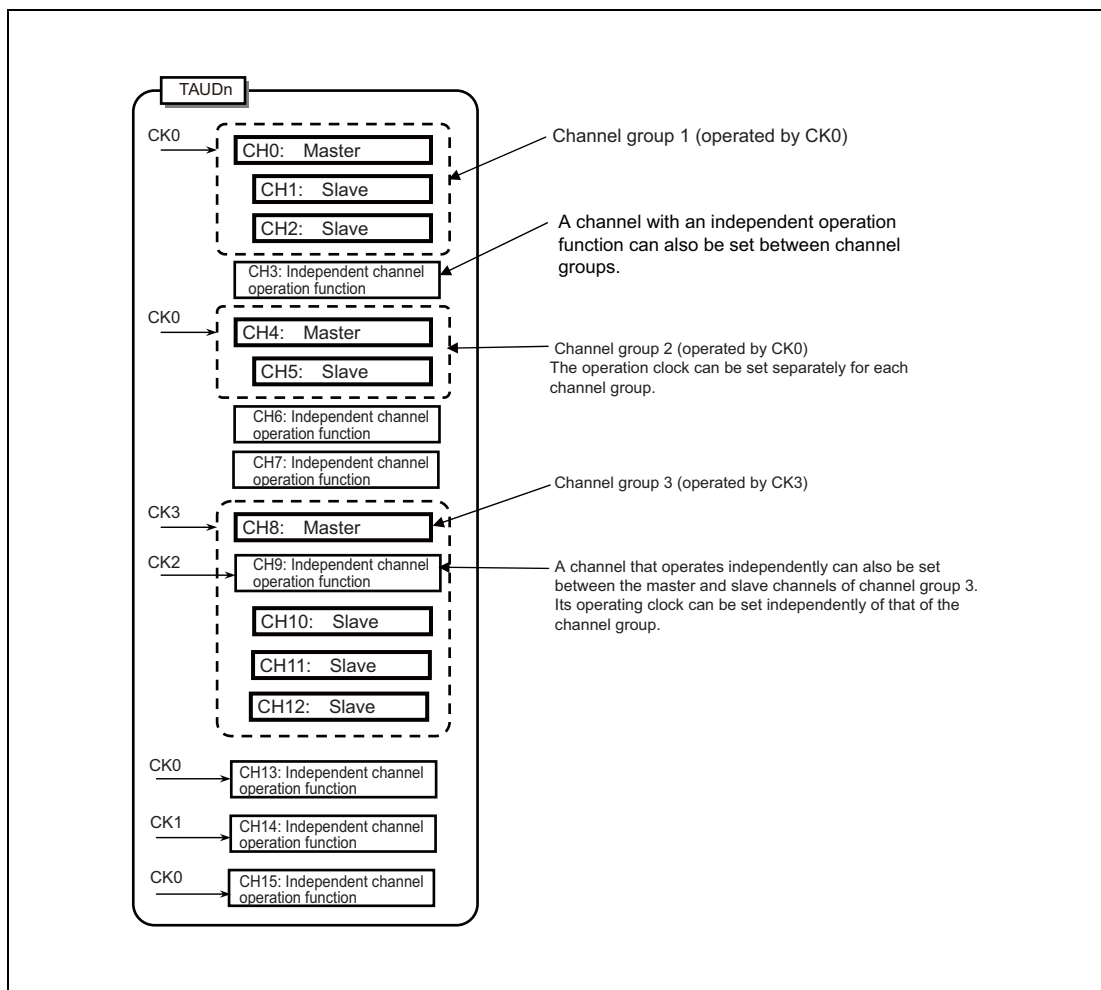


Figure 23.3 Grouping of Channels and Assignment of Operating Clocks

Control trigger signal for master/slave channels

- Master channels can output control trigger signals to slave channels.
- Slave channels can use control trigger signals from master channels but cannot output control trigger signals of their own to lower channels.
- Master channels cannot use control trigger signals from upper master channels.

23.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

23.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUDnTS.TAUDnTSM bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUDnTT.TAUDnTTM bits of the channels should be set at the same time.

Setting to the TAUDnTS.TAUDnTSM bits to 1 also sets the corresponding TAUDnTE.TAUDnTEM bits to 1, enabling counting. The count start timing depends on operating mode.

23.5.2.2 Simultaneous Start between the Units

Counters in different units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details on simultaneous start between the units and other timer functions, **Section 29, Peripheral Interconnection (PIC), 29.2.3.1, Simultaneous Start Trigger Function.**

23.6 Simultaneous Rewrite

23.6.1 Overview of Operations

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUDnCDRm and TAUDnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUDnIm being issued on the upper channel specified by TAUDnRDC.TAUDnRDCm

There are four methods for simultaneous rewrite. These are listed in **Table 23.41**, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 23.41 Simultaneous Rewrite Methods and when They are Triggered

| Method | Simultaneous Rewrite Triggered when | TAUDnRDE. TAUDnRDEm | TAUDnRDS. TAUDnRDSm | TAUDnRDM. TAUDnRDMm |
|--------|---|------------------------|------------------------|------------------------|
| — | There is no simultaneous rewrite | 0 | 0 | 0 |
| A | The master channel (re)starts counting | 1 | 0 | 0 |
| B | The master channel starts counting down at the peaks of the triangular waves in the corresponding slave channel. | 1 | 0 | 1 |
| C1 | INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm | 1 | 1 | 0/1 |
| C2 | INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm that in turn is triggered by an external signal | 1 | 1 | 0/1 |

Table 23.42 lists which of these four methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 23.12, Independent Channel Operation Functions**, **Section 23.13, Synchronous Channel Operation Functions** and **Section 23.14, Synchronous Non-Complementary and Complementary Modulation Output Functions**.

Table 23.42 Channel Functions and the Methods They Use for Simultaneous Rewrite

| Function | A | B | C1 | C2 | TAUDnTOL. TAUDnTOLm |
|---|---|---|----|----|------------------------|
| Simultaneous Rewrite Trigger Output Function Type 1 | | | X | | |
| PWM Output Function | X | | X | | X |
| One-Shot Pulse Output Function | X | | | | |
| Trigger Start PWM Output Function | X | | | X | |
| Delay Pulse Output Function | X | | | | |
| Triangle PWM Output Function | | X | X | | X |
| Triangle PWM Output Function with Dead Time | | X | X | | |
| Interrupt Request Signals Culling Function | X | X | X | | |
| A/D Conversion Trigger Output Function Type 1 | X | | X | | |
| A/D Conversion Trigger Output Function Type 2 | | X | X | | |
| Non-Complementary Modulation Output Function Type 1 | X | | X | | |
| Non-Complementary Modulation Output Function Type 2 | | X | X | | |
| Complementary Modulation Output Function | | X | X | | |

Note: X: Available, (Blank): Unavailable

23.6.2 How to Control Simultaneous Rewrite

Figure 23.4 shows the general procedure for simultaneous rewrite. The three main blocks (initial settings, start and counter count operation, and simultaneous rewrite) are explained afterwards.

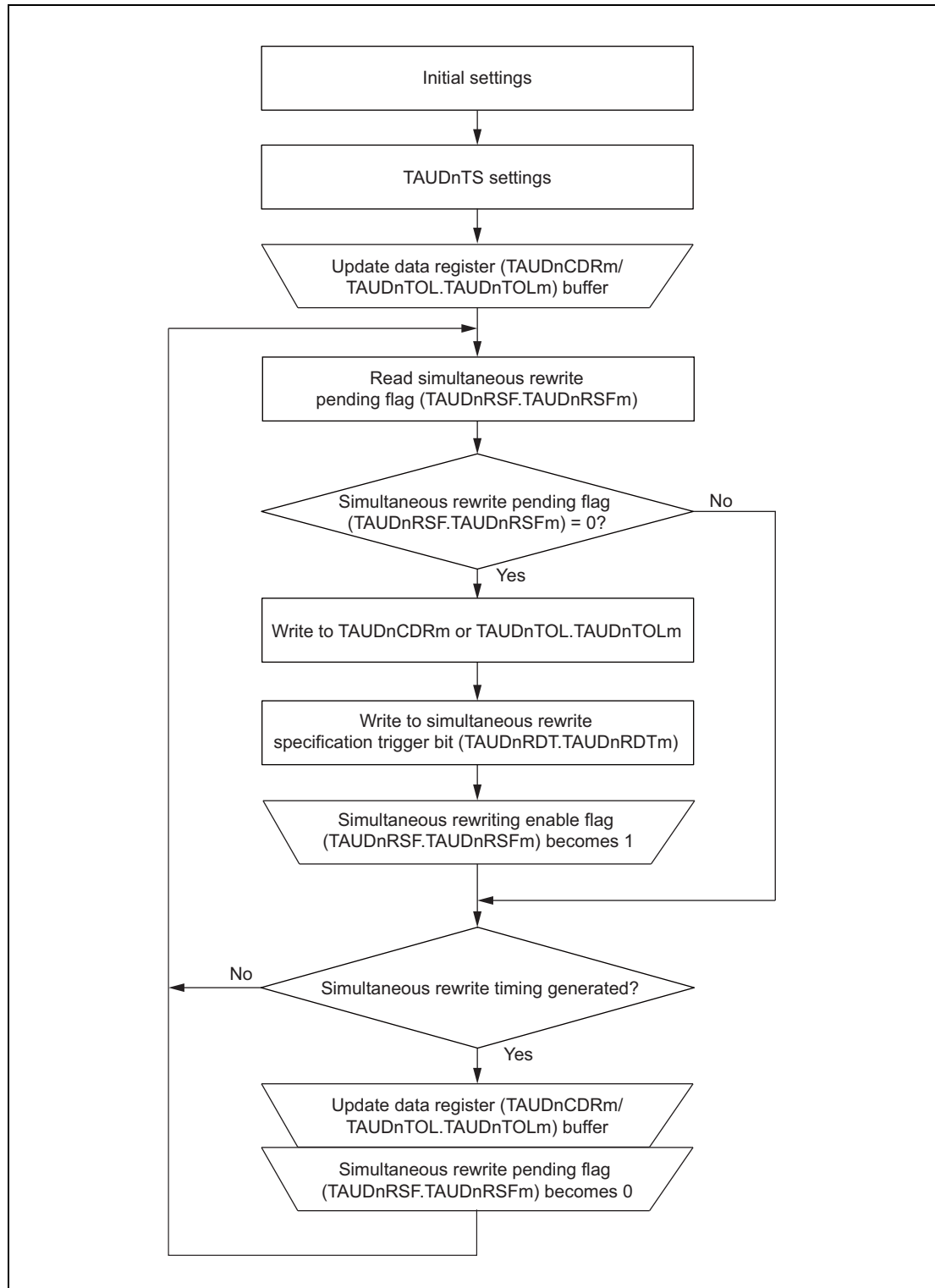


Figure 23.4 General Procedure for Simultaneous Rewrite

23.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set $TAUDnRDE.TAUDnRDEm = 1$
- To select the type of simultaneous rewrite, set $TAUDnRDM.TAUDnRDMm$ and $TAUDnRDS.TAUDnRDSm$ according to the values listed in **Table 23.41, Simultaneous Rewrite Methods and when They are Triggered**.
- Specify a simultaneous rewrite trigger channel by using $TAUDnRDC.TAUDnRDCm$.
(Prerequisite: $TAUDnRDS.TAUDnRDSm$ has been set to the upper channel.)

23.6.2.2 Start Counter and Count Operation

- To start all the $TAUDnCNTm$ counters of the channel group, set the corresponding $TAUDnTS.TAUDnTSm$ bits to 1. The values of $TAUDnTOL.TAUDnTOLm$ and the data registers ($TAUDnCDRm$) are loaded into the corresponding $TAUDnTOL.TAUDnTOLm$ buffer ($TAUDnTOL.TAUDnTOLm$ buf) and data buffer registers ($TAUDnCDRm$ buf) and the counters start.
- Setting the reload data trigger bit ($TAUDnRDT.TAUDnRDTm$) to 1 sets the reload flag ($TAUDnRSF.TAUDnRSFm$) to 1, enabling simultaneous rewrite. $TAUDnRSF.TAUDnRSFm$ remains set to 1 until simultaneous rewrite is completed.
- When the specified trigger for simultaneous rewrite is detected, the $TAUDnRSF.TAUDnRSFm$ bit is checked to see if simultaneous rewrite is enabled ($TAUDnRSF.TAUDnRSFm = 1$). If it is, simultaneous rewrite is carried out. Otherwise the simultaneous rewrite is not carried out and waits for the next trigger detection.

23.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled ($TAUDnRSF.TAUDnRSFm = 1$) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and are applied the next time the counter starts or restarts.
- When simultaneous rewrite is complete, the $TAUDnRSF.TAUDnRSFm$ bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

23.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUDnRDE.TAUDnRDEm, TAUDnRDS.TAUDnRDSm, TAUDnRDM.TAUDnRDMm, and TAUDnRDC.TAUDnRDCm cannot be changed while the counter is in operation (TAUDnTE.TAUDnTEm = 1).
- TAUDnTOL.TAUDnTOLm can only be rewritten during operation with PWM output function or triangle PWM output function. For all other output functions, TAUDnTOL.TAUDnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUDnTTOUTm outputs an invalid wave.
- When an upper channel is used as a channel issuing the simultaneous rewrite trigger (TAUDnRDS.TAUDnRDSm = 1), the TAUDnRDC.TAUDnRDCm bit controls all the lower channels. This means that if the TAUDnRDC.TAUDnRDCm bits of CH2 and CH7 are set to 1 and the TAUDnRDC.TAUDnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected as the channel to generate the simultaneous rewrite trigger (TAUDnRDE.TAUDnRDEm and TAUDnRDS.TAUDnRDSm = 1) but no upper channel is set (TAUDnRDC.TAUDnRDC[15:0] = 0), simultaneous rewrite cannot take place.

23.6.4 Types of Simultaneous Rewrite

In the following section, the four simultaneous rewrite methods are explained using timing diagrams.

23.6.4.1 Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)

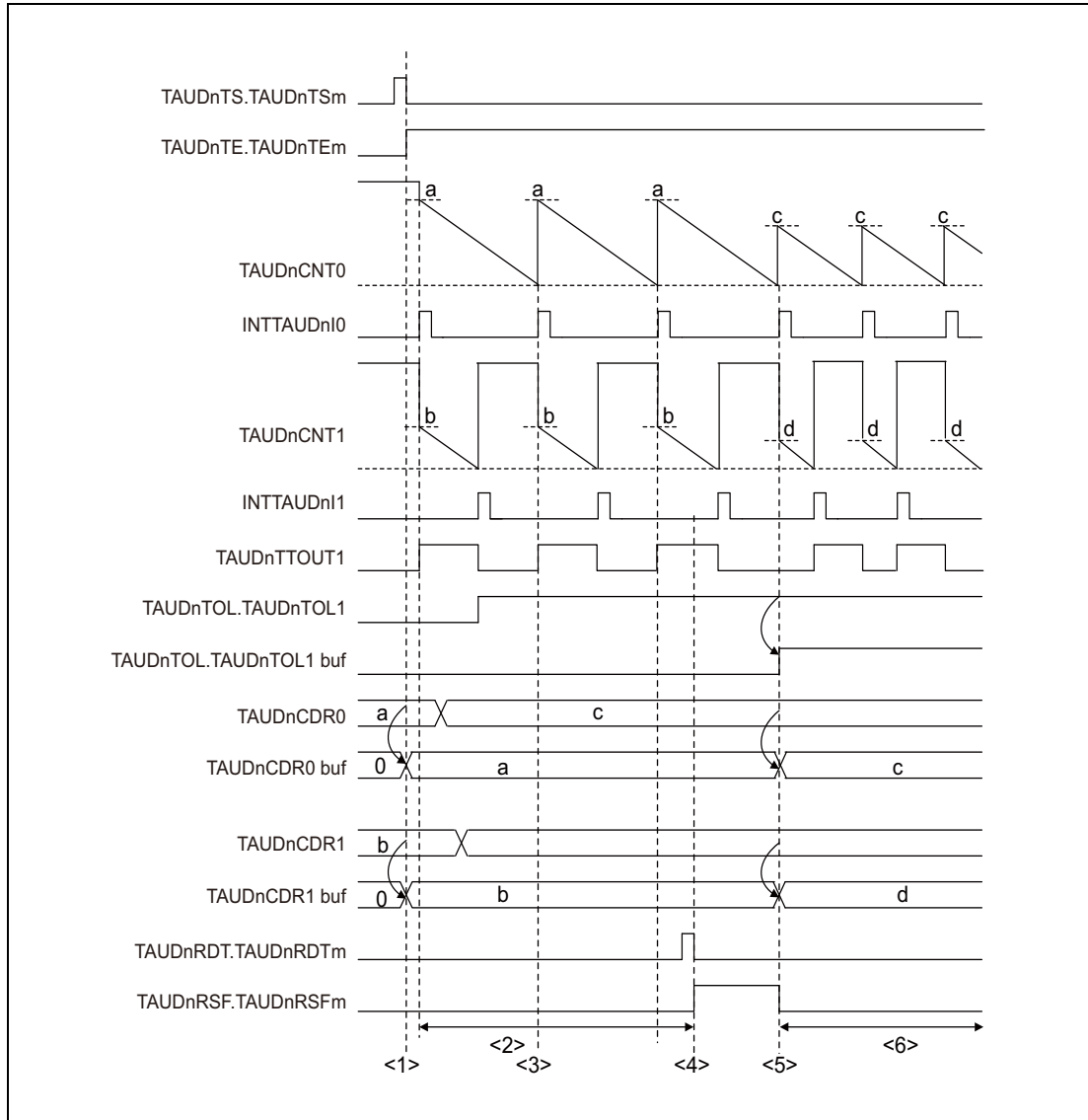


Figure 23.5 Simultaneous Rewrite when the Master Channel (Re)starts Counting

Setup:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

- (1) When $\text{TAUDnTS.TAUDnTSM} = 1$ is set, the value of TAUDnCDRm is copied to the TAUDnCDRm buffer and the value of $\text{TAUDnTOL.TAUDnTOLm}$ is copied to the $\text{TAUDnTOL.TAUDnTOLm}$ buffer.
- (2) The TAUDnCDRm and $\text{TAUDnTOL.TAUDnTOLm}$ registers can be written at any time.
- (3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled ($\text{TAUDnRSF.TAUDnRSFm} = 0$).
- (4) The reload data trigger bit ($\text{TAUDnRDT.TAUDnRDTm}$) is set to 1 which sets the status flag ($\text{TAUDnRSF.TAUDnRSFm} = 1$), enabling simultaneous rewrite.
- (5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer and the $\text{TAUDnTOL.TAUDnTOLm}$ value is loaded into the $\text{TAUDnTOL.TAUDnTOLm}$ buffer.
- (6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and $\text{TAUDnTOL.TAUDnTOLm}$ can be changed again.

23.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B)

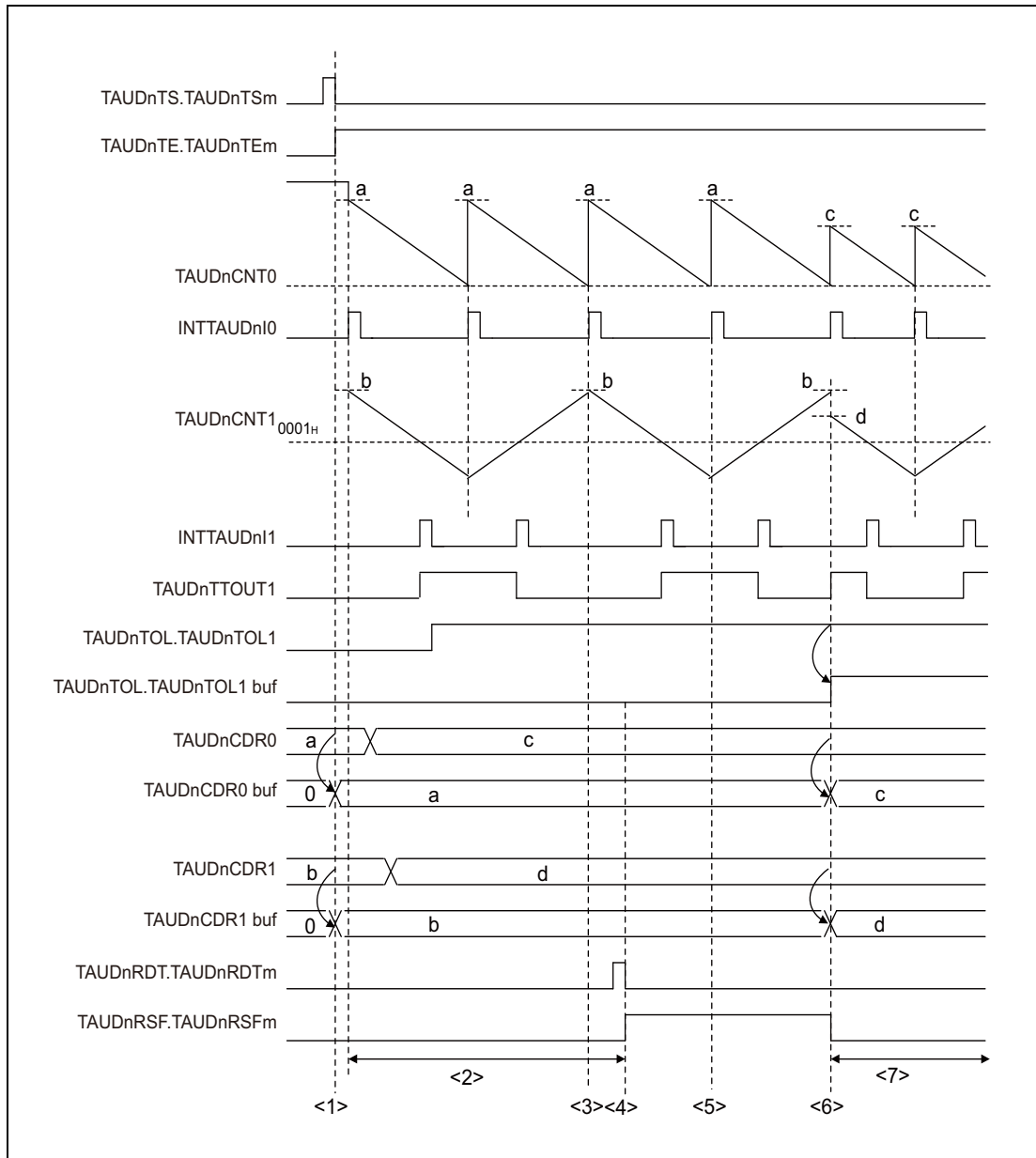


Figure 23.6 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel

Setting

CH0 is the master channel which performs counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

- (1) When $\text{TAUDnTS.TAUDnTSM} = 1$ is set, the value of TAUDnCDRm is copied to the TAUDnCDRm buffer.
- (2) The TAUDnCDRm and TAUDnTOL registers can be written at any time.
- (3) Simultaneous rewrite does not occur because it is disabled ($\text{TAUDnRSF.TAUDnRSFm} = 0$).
- (4) The reload data trigger bit ($\text{TAUDnRDT.TAUDnRDTm}$) is set to 1 which sets the status flag ($\text{TAUDnRSF.TAUDnRSFm} = 1$), enabling simultaneous rewrite.
- (5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.
- (6) Simultaneous rewrite takes place at the top of the triangular cycle. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer, the $\text{TAUDnTOL.TAUDnTOLm}$ value is loaded into the $\text{TAUDnTOL.TAUDnTOLm}$ buffer.
- (7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and $\text{TAUDnTOL.TAUDnTOLm}$ can be changed again.

23.6.4.3 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1)

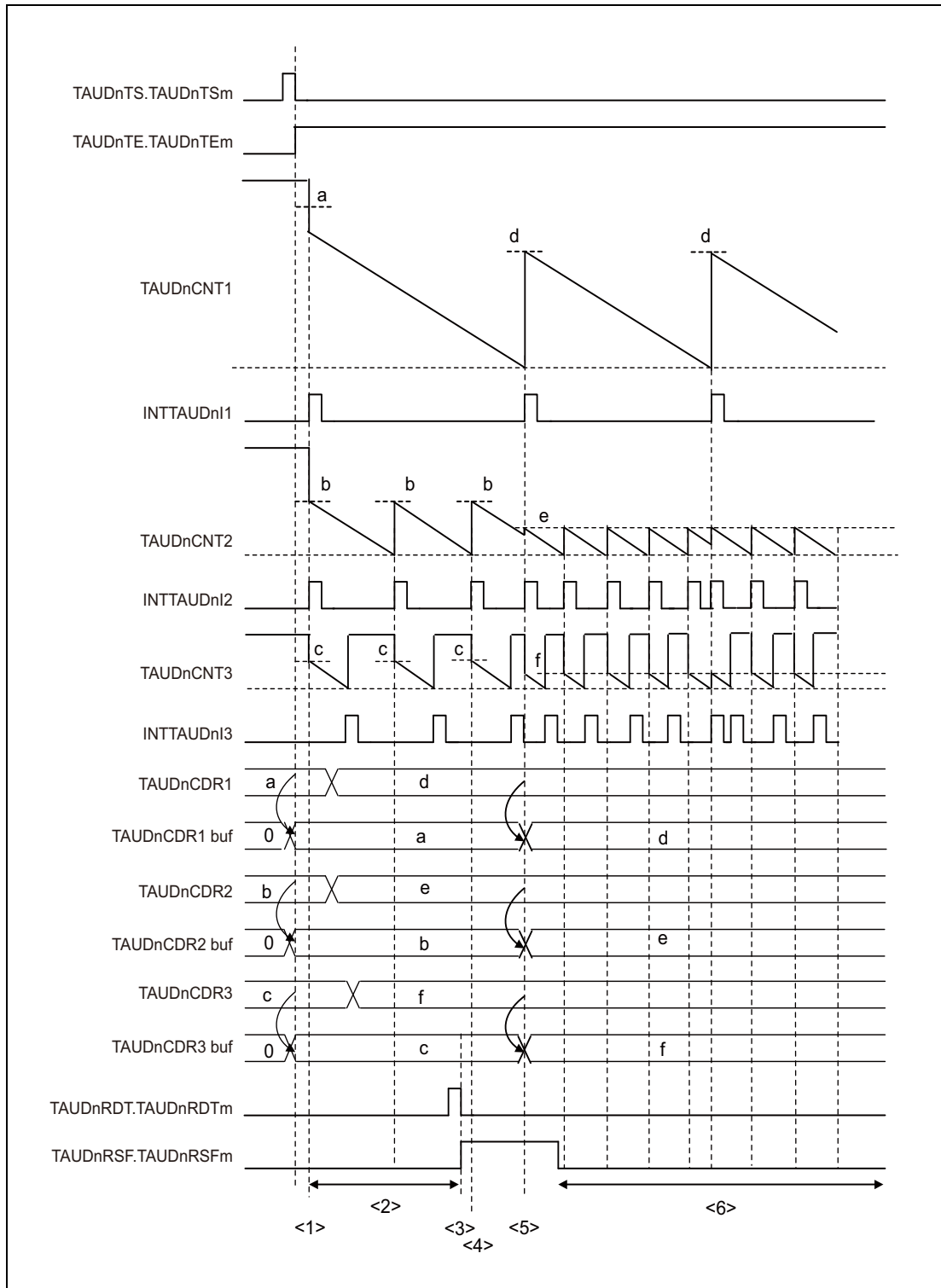


Figure 23.7 Simultaneous Rewrite When INTTAUDnIm Is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm

Setting

CH1 is an upper channel used counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUDnRDC register specifies a channel which generates simultaneous rewrite triggers.

Description:

- (1) When TAUDnTS.TAUDnTSM is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.
- (2) The TAUDnCDRm register is always ready to write.
- (3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.
- (4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.
- (5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000_H. The TAUDnCDRm values are loaded into the corresponding TAUDnCDRm buffers.
- (6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be changed again.

23.6.4.4 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal (Method C2)

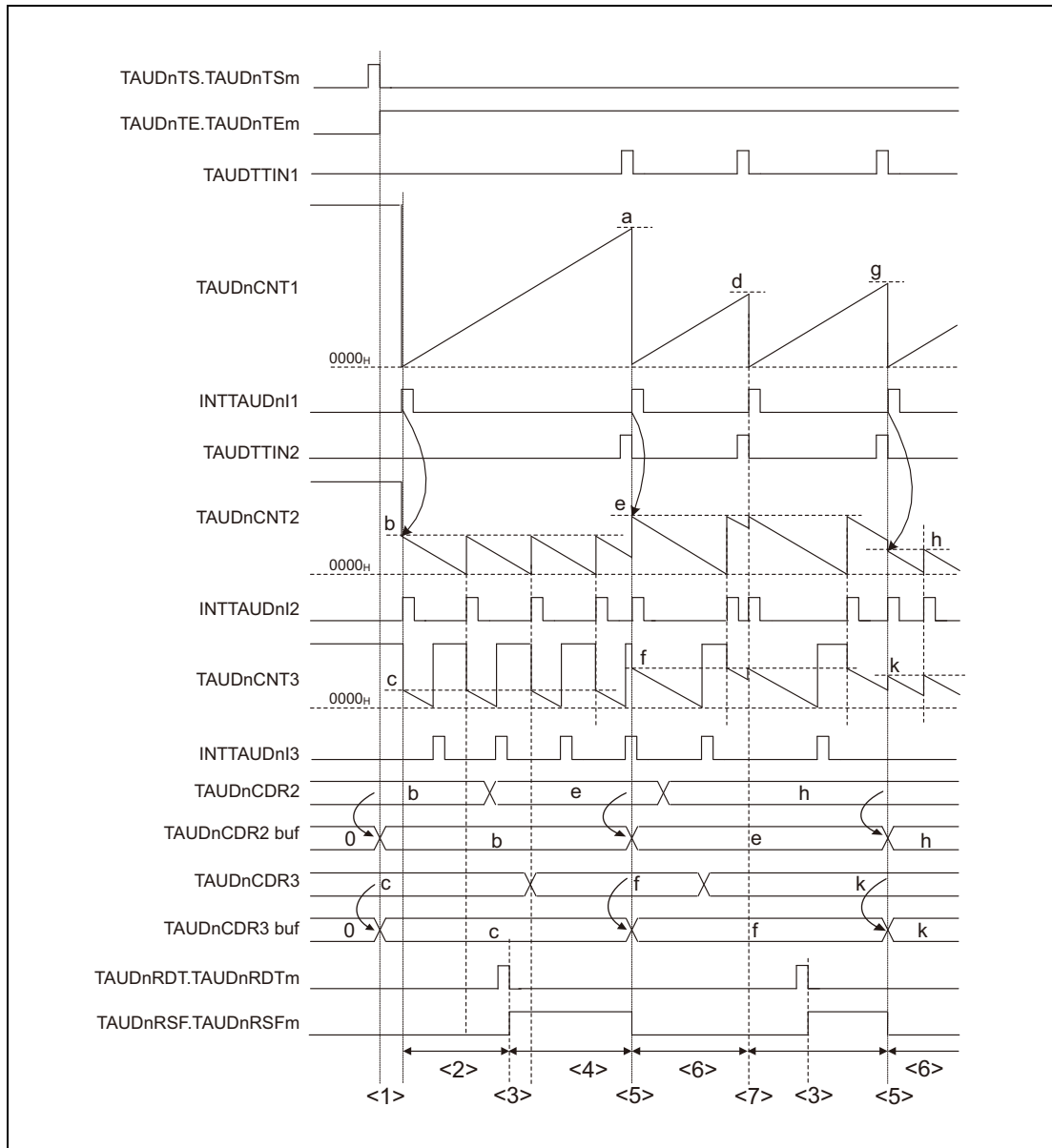


Figure 23.8 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal

Setup:

CH1 is an upper channel used counting up, CH2 is a master channel, and CH3 is the slave channel. The synchronous channel operation method C2 is applied. The TAUDnRDC register specifies which upper channel is monitored for an INTTAUDnIm trigger.

Description:

- (1) When TAUDnTS.TAUDnTSM is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer. However, the TAUDnCDR1 value is not copied to the TAUDnCDR1 buffer because TAUDnCDR1 is in capture mode.
- (2) The TAUDnCDRm register is always ready to write.
- (3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.
- (4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.
- (5) Simultaneous rewrite is triggered by INT1 which is caused by external signal TIN1. The TAUDnCDRm values are written to the corresponding TAUDnCDRm buffers.
- (6) The counters count down and await the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be changed again.
- (7) An external signal occurs at TIN2 but simultaneous rewrite does not take place because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

23.7 Channel Output Modes

The output of the TAUDnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUDnTOE.TAUDnTOEm = 0)
When controlled by software, the value written in the output register bit (TAUDnTO.TAUDnTOm) is sent out of the output pin (TAUDnTTOUTm).
- By TAUD signals (TAUDnTOE.TAUDnTOEm = 1)
When controlled by TAUD signals, the output level of TAUDnTTOUTm is set or reset or toggled by internal signals. The value of TAUDnTO.TAUDnTOm is updated accordingly to reflect the value of TAUDnTTOUTm
 - Independently (TAUDnTOM.TAUDnTOMm = 0)
In case of independent operation, the output of the TAUDnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUDnTOM.TAUDnTOMm = 0).
 - Synchronously (TAUDnTOM.TAUDnTOMm = 1)
In case of synchronous operation, the output of the TAUDnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUDnTOM.TAUDnTOMm = 1).

The TAUDnTO.TAUDnTOm bit can always be read to determine the current value of TAUDnTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 23.43, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 23.7.2, Channel Output Modes Controlled Independently by TAUDn Signals**
- **Section 23.7.3, Channel Output Modes Controlled Synchronously by TAUDn Signals**

Batch operation of TAUDnTOM bit

Whether a set value is reflected to the TAUDnTOM bit or not is controlled by the TAUDnTOE.TAUDnTOEm bit.

The TAUDnTOM setting is written only to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 0 when a write to the TAUDnTO register is attempted. No TAUDnTOM setting is reflected to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 1.

NOTE

TAUDnTO.TAUDnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUDnTOL.TAUDnTOLm.

The value of TAUDnTOL.TAUDnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. Otherwise, changes to TAUDnTOL.TAUDnTOLm result in an invalid TAUDnTTOUTm signal output.

See **Section 23.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 23.43**.

Table 23.43 Channel Output Modes

| Channel Output Mode | TAUDn TOE. TAUDn TOEm | TAUDn TOM. TAUDn TOMm | TAUDn TOC. TAUDn TOCm | TAUDn TDE. TAUDn TDEm | TAUDn TRE. TAUDn TREM | TAUDn TME. TAUDn TMEm | TAUDn TDM. TAUDn TDMm | | |
|--|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|---|---|
| By software | | | | | | | | | |
| Independent channel output mode controlled by software | 0 | | | | | X | | | |
| By TAUD signals, independently | | | | | | | | | |
| Independent channel output mode 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| with real-time output | | | | | 1 | | | | |
| Independent channel output mode 2 | | | 1 | | 0 | | | | |
| By TAUD signals, synchronously | | | | | | | | | |
| Synchronous channel output mode 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | |
| with non-complementary modulation output | | | | | 1 | | | X | |
| Synchronous channel output mode 2 | | | 1 | | 0 | | | 0 | 0 |
| with dead time output | | | | | 1 | | | | |
| with one-phase PWM output | | | | | | | | | |
| with complementary modulation output | | | | | | | | | 1 |
| with non-complementary modulation output | | | 1 | 0 | | | | | |

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTES

1. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1):
 - TAUDnTOM.TAUDnTOMm
 - TAUDnTOC.TAUDnTOCm
 - TAUDnTDE.TAUDnTDEm
 - TAUDnTRE.TAUDnTREm
 - TAUDnTDM.TAUDnTDMm
2. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1) except in channel output modes with modulation output:
 - TAUDnTME.TAUDnTMEm
 - TAUDnTDL.TAUDnTDLm

23.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUDnTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUDnTOE.TAUDnTOEm = 0).

- (1) Set TAUDnTO.TAUDnTOm to specify the initial level of the TAUDnTTOUTm output.
- (2) Set channel output mode according to **Table 23.43, Channel Output Modes**, and the output logic using the TAUDnTOL.TAUDnTOLm bit.
- (3) Start the counter (TAUDnTS.TAUDnTSM = 1).

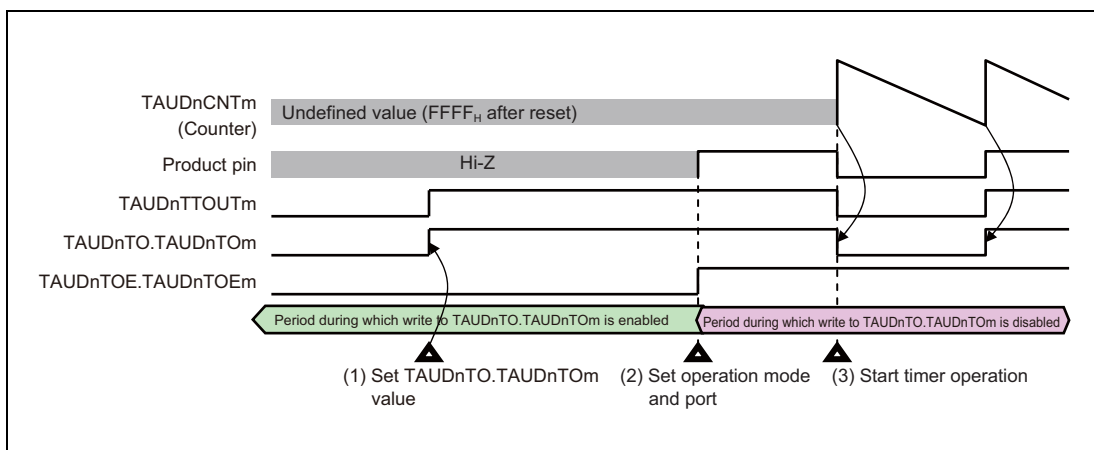


Figure 23.9 General Procedure for Specifying a TAUDnTTOUTm Channel Output Mode

23.7.2 Channel Output Modes Controlled Independently by TAUDn Signals

This section lists the channel output modes that are controlled independently by TAUDn signals. The control bits used to specify a mode are listed in **Table 23.43, Channel Output Modes**.

23.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUDnTTOUTm toggles when INTTAUDnIm is detected. The value of TAUDnTOL.TAUDnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 23.43, Channel Output Modes**

23.7.2.2 Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of TAUDnTRO.TAUDnTROm bit of the trigger channel is output to TAUDnTTOUTm. The trigger channel is specified by setting the corresponding TAUDnTRC.TAUDnTRCm bit to 1. It controls all lower channels for which TAUDnTRC.TAUDnTRCm = 0.

Set/reset conditions

The value of TAUDnTRO.TAUDnTROm bit is sent to TAUDnTTOUTm only when an INTTAUDnIm interrupt occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of a valid TAUDnTTINm input edge/counter start

The type of trigger is set using the TAUDnCMORm.TAUDnMD[4:1] bits.

Prerequisites

Both master and slave channels can be set as a trigger generation channel. A channel for which TAUDnTRC.TAUDnTRCm is set to 1 serves as a trigger generation channel regardless of the value of TAUDnTRE.TAUDnTREM.

If there is no channel for which TAUDnTRC.TAUDnTRCm is set to 1 or if TAUDnTRC.TAUDnTRC0 = 0, real-time output cannot take place.

This can be seen in **Figure 23.10**.

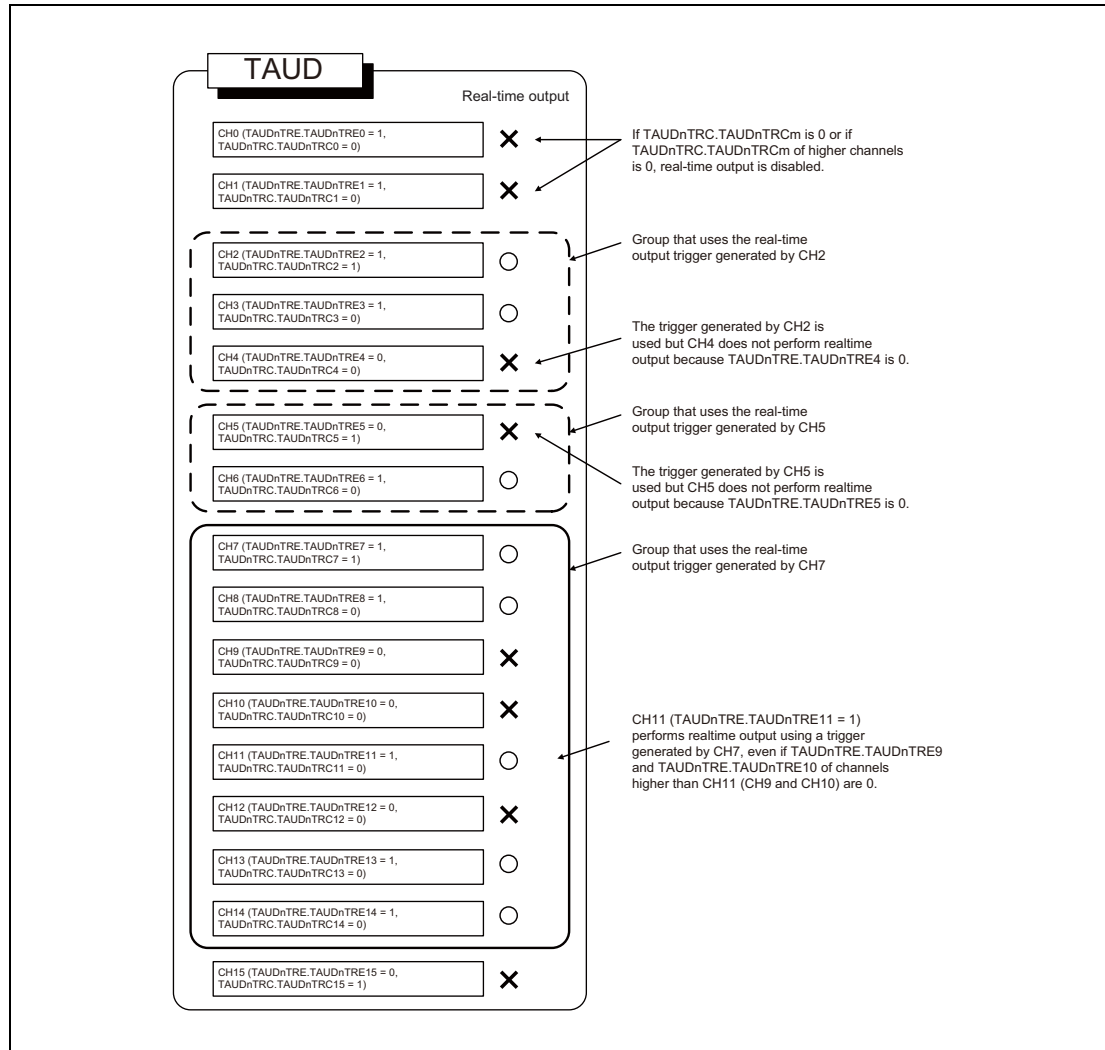


Figure 23.10 Real-Time Output

23.7.2.3 Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUDnTTOUTm is set when INTTAUDnIm occurs at the time of count start, and reset when INTTAUDnIm occurs due to a match between TAUDnCNTm and TAUDnCDRm.

Prerequisites

There are no prerequisites other than those shown in **Table 23.43, Channel Output Modes**.

23.7.3 Channel Output Modes Controlled Synchronously by TAUDn Signals

This section lists the channel output modes that are controlled synchronously by TAUDn signals. The control bits used to specify a mode are listed in **Table 23.43, Channel Output Modes**.

23.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUDnIm of master channel serves as a set signal and INTTAUDnIm of the slave channel as a reset signal. If INTTAUDnIm of master channel and INTTAUDnIm of the slave channel are generated at the same time, INTTAUDnIm of the slave channel (reset signal) has priority over INTTAUDnIm (set signal) of master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 23.43, Channel Output Modes**.

23.7.3.2 Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Set/reset conditions

In this output mode, TAUDnTTOUTm outputs the result of an AND operation between the PWM output and the real-time output bit (TAUDnTRO.TAUDnTROM) of a channel.

The phase period to which the dead time is added is specified using the TAUDnTDL.TAUDnTDLm bit; for positive phase set TAUDnTDL.TAUDnTDLm = 0 and for negative phase set TAUDnTDL.TAUDnTDLm = 1.

Prerequisites

A set of at least three channels is required to generate the PWM output. The master channel and slave channel 1 generate a period, and slave channel 2 generates the duty cycle. In typical applications, five more slave channels are also used that operate in the same manner as slave channel 2.

Only the PWM output and the real-time output bit of the same channel can be combined.

TAUDnTRO.TAUDnTROM, TAUDnTME.TAUDnTME m, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTME m is changed, its new value is applied upon detection of INTTAUDnIm on the specified channel.
- If TAUDnTME.TAUDnTME m and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on the master channel.

23.7.3.3 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to up/down count mode. The result is a triangle PWM output at $TAUDnTTOUTm$. For details, see **Section 23.13.6, Triangle PWM Output Function**.

Set/reset conditions

$TAUDnCNTm$ of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing $TAUDnTTOUTm$ to toggle.

Prerequisites

A set of two channels is required to generate the triangle PWM output. $TAUDnTTOUTm$ should be set to 0 before the function starts.

23.7.3.4 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to $TAUDnTTOUTm$. The set/reset conditions are shown in **Figure 23.11**.

Set/reset conditions

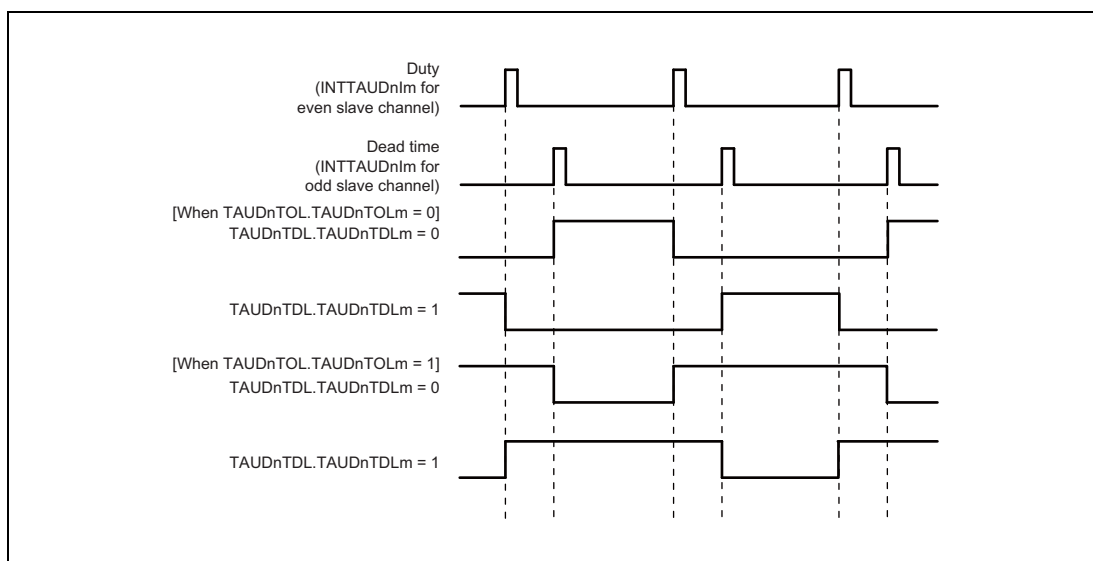


Figure 23.11 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set $TAUDnTDL.TAUDnTDLm = 0$ for rising edges and $TAUDnTDL.TAUDnTDLm = 1$ for falling edges.

Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel
The master channel should be set to interval timer mode.
- One even-numbered slave channel
The even-numbered slave channel should be set up/down count mode.
- One odd-numbered slave channel (even-numbered channel + 1)
The odd-numbered slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd-numbered channel and the even-numbered channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTME_m
- TAUDnTRE.TAUDnTRE_m
- TAUDnTOM.TAUDnTOM_m
- TAUDnTOC.TAUDnTOC_m
- TAUDnTDE.TAUDnTDE_m
- TAUDnTDM.TAUDnTDM_m

23.7.3.5 Synchronous Channel Output Mode 2 with One-Phase PWM Output

In this output mode, a dead time delay is added to TAUDnTTOUT_m. The set/reset conditions are shown in **Figure 23.12**.

Set/reset conditions

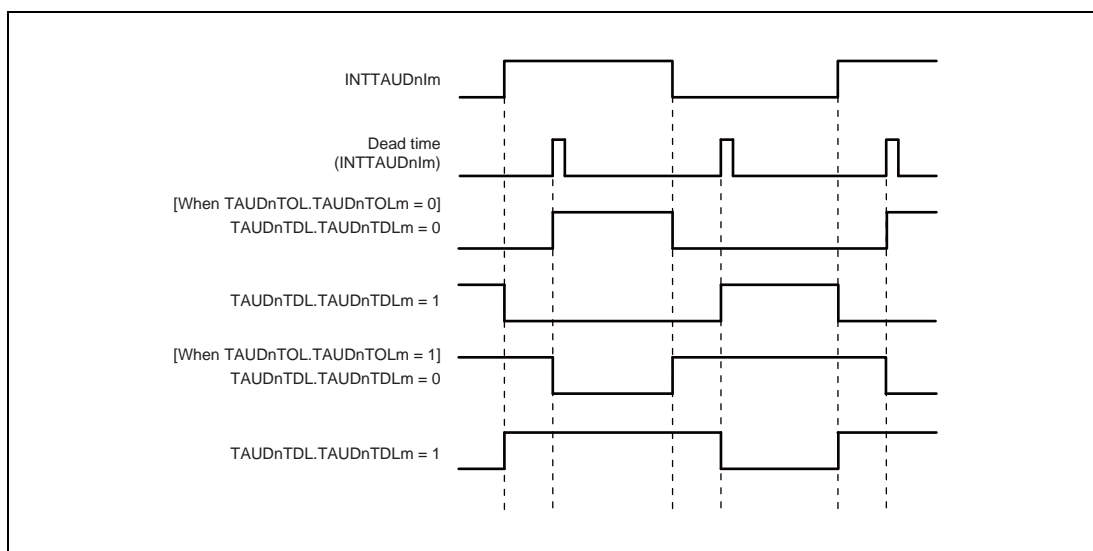


Figure 23.12 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDL_m = 0 for rising edges and TAUDnTDL.TAUDnTDL_m = 1 for falling edges.

Prerequisites

One-phase PWM output control requires a set of two channels:

- One even-numbered slave channel
- One odd-numbered slave channel (even-numbered channel + 1)
The odd-numbered slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd-numbered channel and the even-numbered channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm
- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

23.7.3.6 Synchronous Channel Output Mode 2 with Complementary Modulation Output**Set/reset conditions**

In this output mode, TAUDnTTOUTm outputs a PWM signal, a high signal, or a low signal depending on the value of real-time output bit (TAUDnTRO.TAUDnTROm), the modulation output bit (TAUDnTME.TAUDnTMEm), and the output level bit (TAUDnTOL.TAUDnTOLm) of a pair of slave channels.

For details, see **Section 23.14.3, Complementary Modulation Output Function**.

Prerequisites

A set of at least four channels is required for this mode. The master channel and slave channel 1 generate a period, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time. Slave channels 2 and 3 are a pair. In typical applications, 4 more channels are also used, which operates in the same manner as slave channels 2 and 3 respectively.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed during operation, its new value is applied upon detection of INTTAUDnIm at the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on an even-numbered slave channel.

23.7.3.7 Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

The difference from synchronous channel output mode 1 with non-complementary modulation output is the PWM wave shape.

Mode 1 has a square wave while mode 2 has a triangular wave.

23.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after `TAUDnTS.TAUDnTSM` is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

23.8.1 Interval Timer Mode, Judge Mode, Capture Mode, and Up/Down Count Mode

The counter starts operating with the next count clock after `TAUDnTS.TAUDnTSM` is set to 1. The value of data register is also loaded when the counter starts.

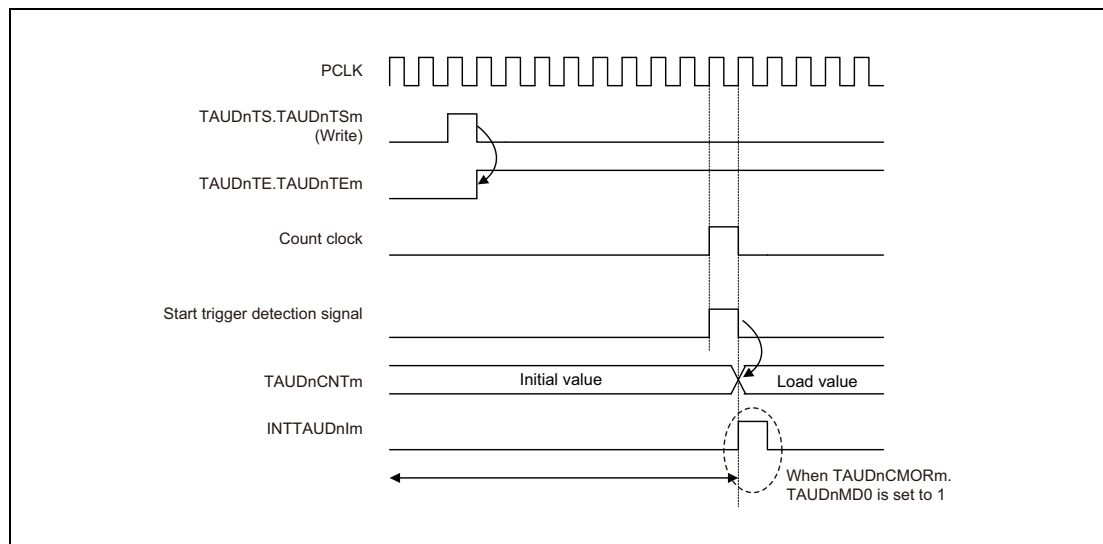


Figure 23.13 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, and Up/Down Count Mode

NOTE

Make sure to set `TAUDnCMORm.TAUDnMD0 = 0` when using the up/down count mode.

23.8.2 Event Count Mode

The value of data register is loaded as soon as TAUDnTS.TAUDnTSM is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

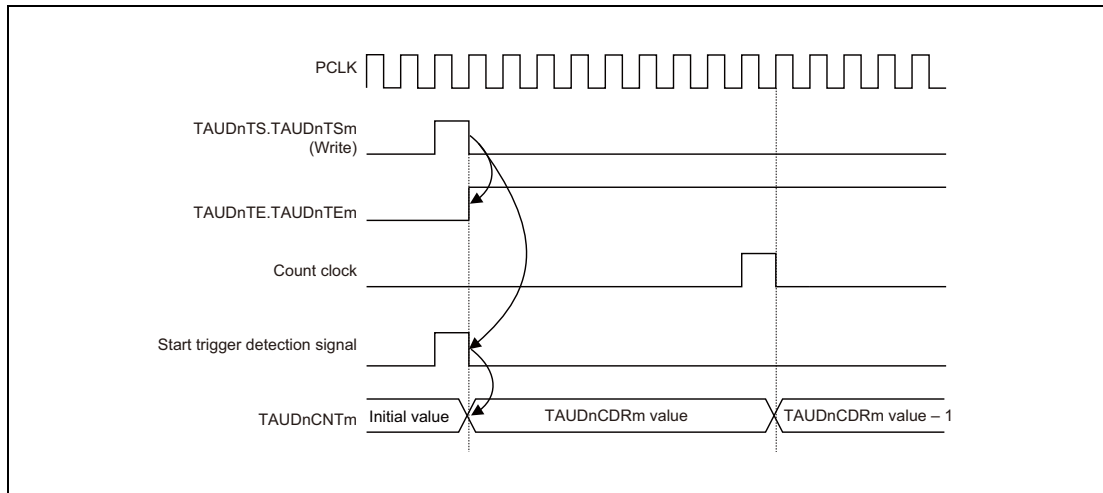


Figure 23.14 Start Timing in Event Count Mode

23.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUDnTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place.

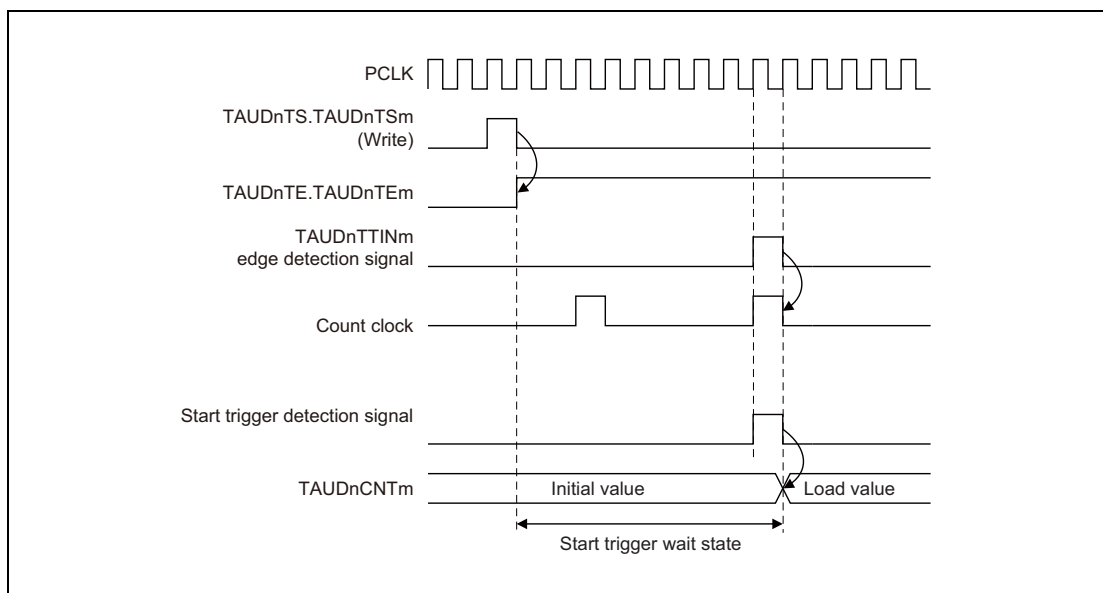


Figure 23.15 Count Start Timing in Other Operating Modes

23.9 TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUDnIm is generated using the TAUDnCMOR.TAUDnMD0 bit. The way in which the setting of the TAUDnCMORm.TAUDnMD0 bit determines whether an INTTAUDnIm interrupt is generated at the start of counting and the operation of TAUDnTTOUTm depends on the selected mode. For details, see the description of TAUDnCMORm.TAUDnMD0 in the relevant individual functional blocks.

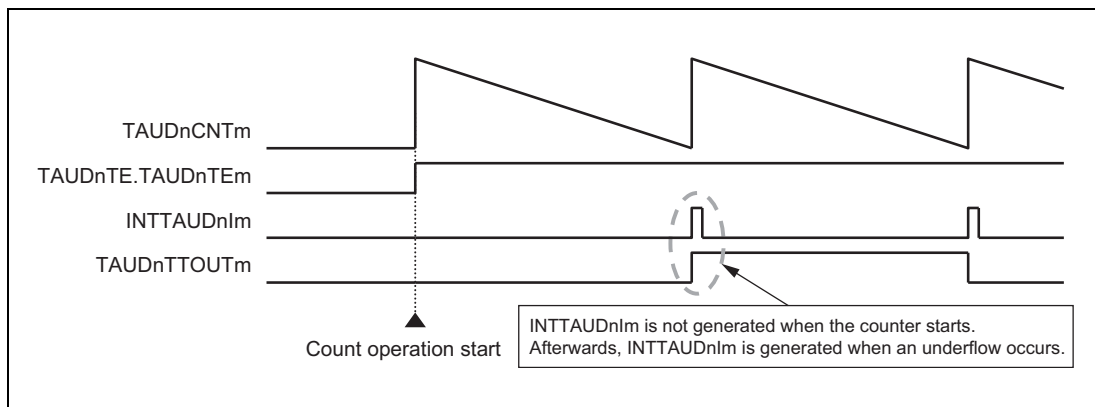


Figure 23.16 INTTAUDnIm Generation Timing (When TAUDnCMORm.TAUDnMD0 = 0)

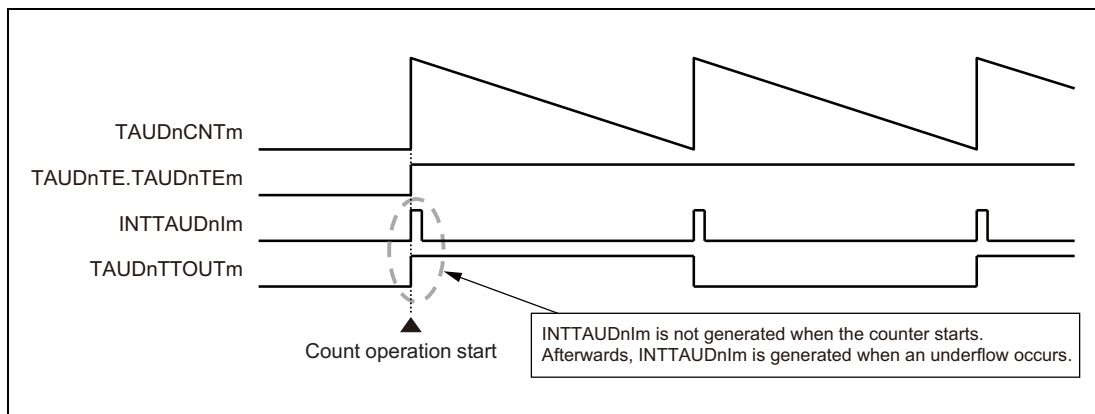


Figure 23.17 INTTAUDnIm Generation Timing (When TAUDnCMORm.TAUDnMD0 = 1)

23.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches $FFFF_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operation in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows ($TAUDnCNTm = FFFF_H$).
- Set $TAUDnCDRm$ of the second channel to $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same $TAUDnTTINm$ input signal.
- The trigger detection settings ($TAUDnCMORm.TAUDnSTS[2:0]$ and $TAUDnCMURm.TAUDnTIS[1:0]$) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows ($TAUDnCNTm = FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

23.10.1 Combination of TAUDTTINm Input Pulse Interval Measuring Function and TAUDTTINm Input Interval Timer Function

When a capture trigger is input to TAUDTTINm of both channels simultaneously, an overflow from FFFF_H in TAUDnCNTm of the TAUDTTINm input pulse interval measuring function is detected by INTTAUDnIm from the TAUDTTINm input interval timer function.

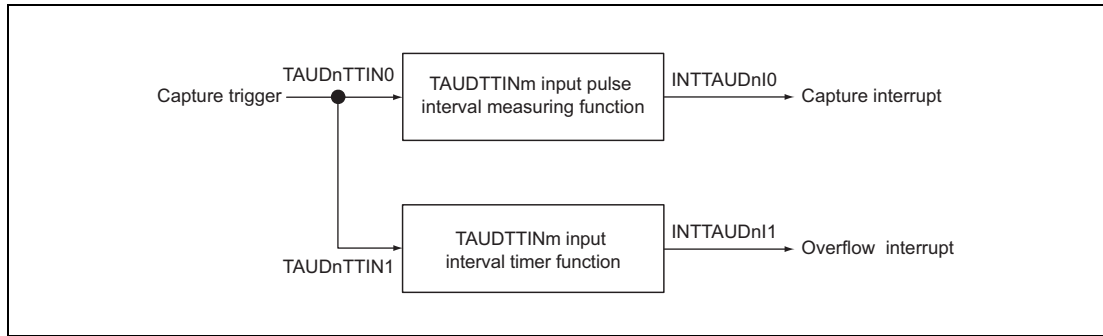


Figure 23.18 Combination of TAUDTTINm Input Pulse Interval Measuring Function and TAUDTTINm Input Interval Timer Function

Timing diagram

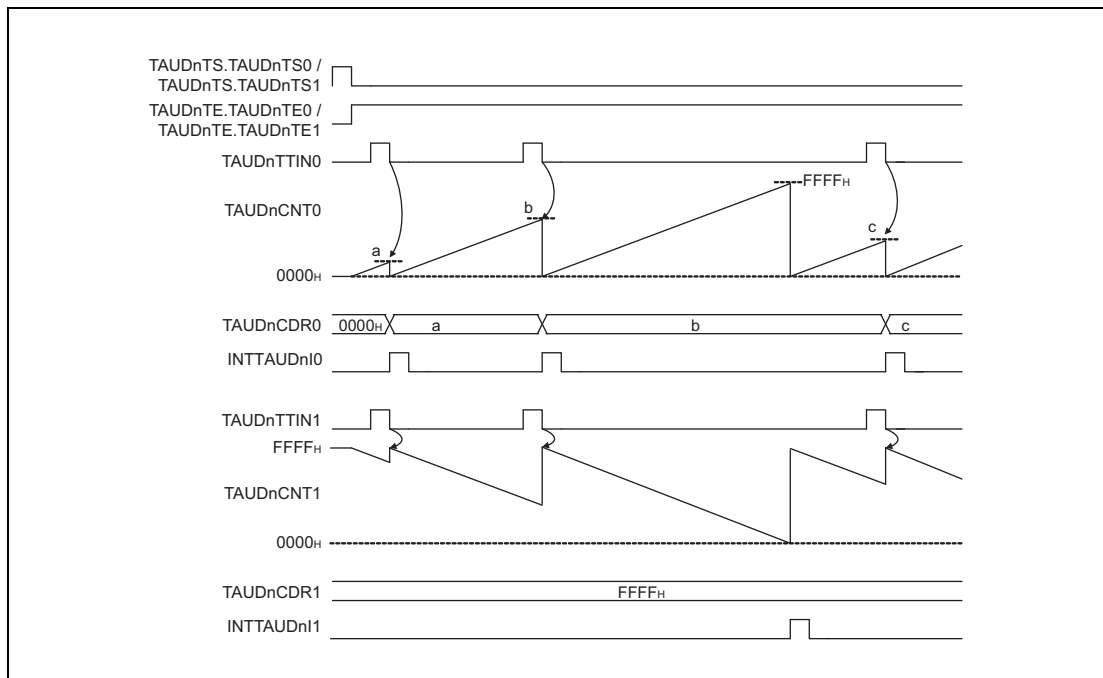


Figure 23.19 Interrupt Generation via Combination of TAUDTTINm Input Pulse Interval Measuring Function and TAUDTTINm Input Interval Timer Function

23.10.2 Combination of TAUDTTINm Input Signal Width Measuring Function and Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

When a capture trigger is input to TAUDTTINm of both channels simultaneously, an overflow from FFFF_H in TAUDnCNTm of the TAUDTTINm input signal width measuring function is detected by INTTAUDnIm from the overflow interrupt output function (during TAUDTTINm width measurement).

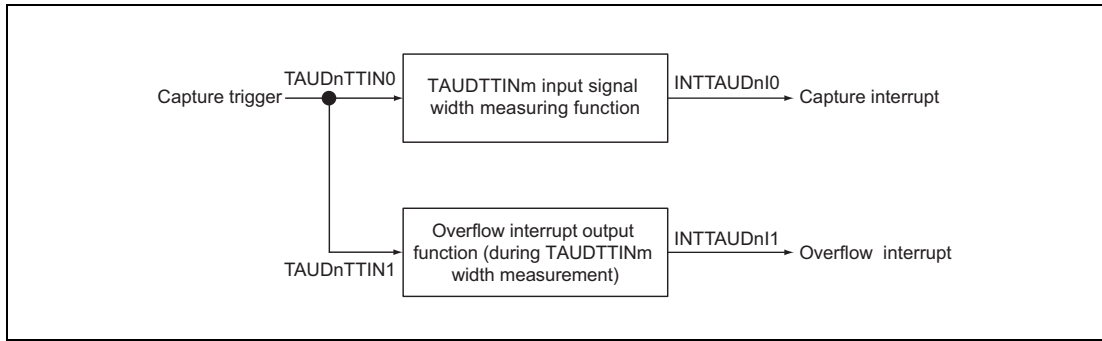


Figure 23.20 Combination of TAUDTTINm Input Signal Width Measuring Function and Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Timing diagram

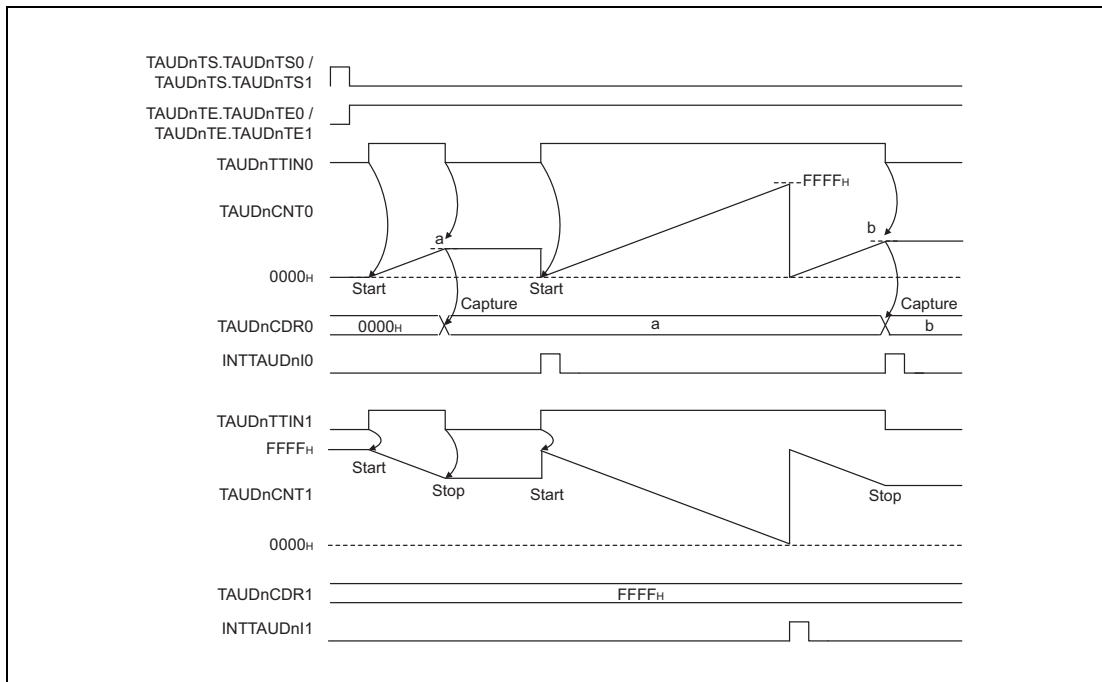


Figure 23.21 Interrupt Generation via Combination of TAUDTTINm Input Signal Width Measuring Function and Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

23.10.3 Combination of TAUDTTINm Input Position Detecting Function and Interval Timer Function

When the counters of both channels are started simultaneously, an overflow from $FFFF_H$ in TAUDnCNTm of the TAUDTTINm input position detecting function is detected by INTTAUDnIm from the interval timer function.

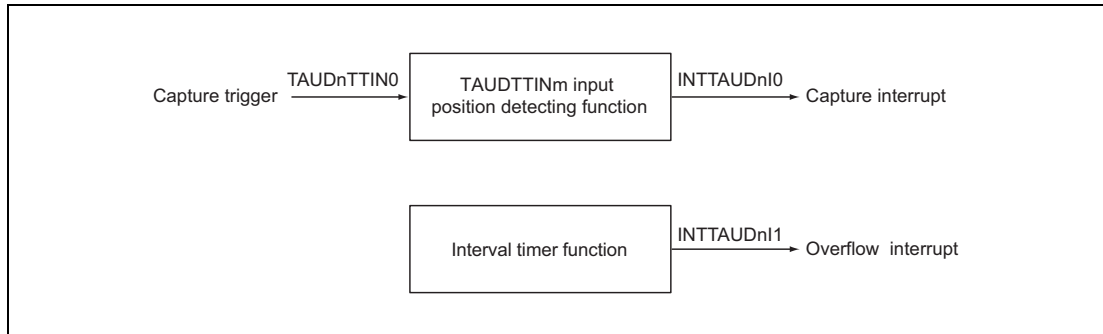


Figure 23.22 Combination of TAUDTTINm Input Position Detecting Function and Interval Timer Function

Timing diagram

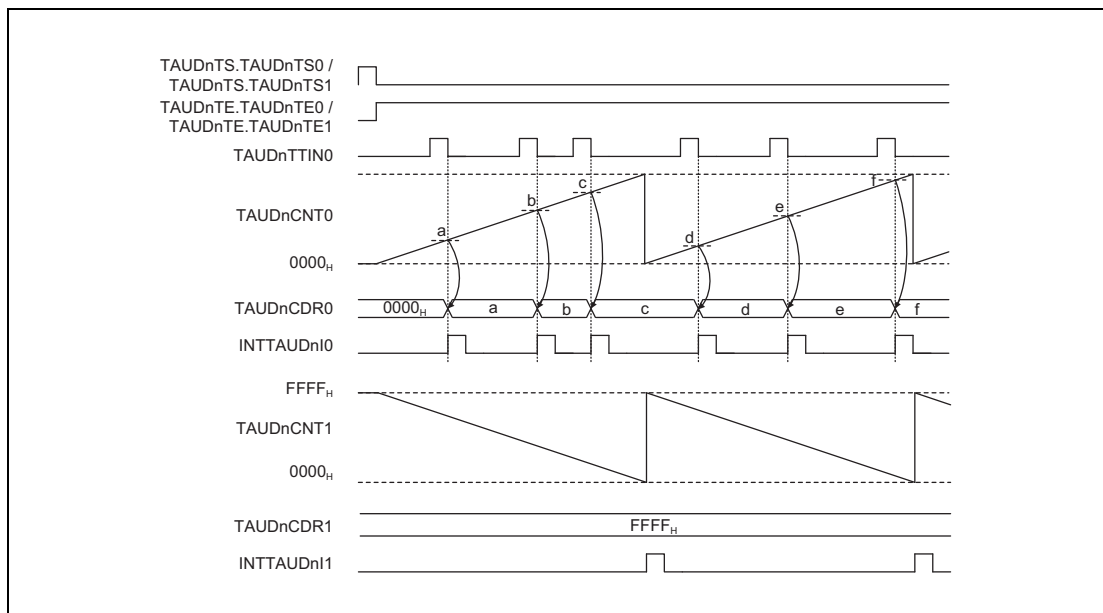


Figure 23.23 Interrupt Generation via Combination of TAUDTTINm Input Position Detecting Function and Interval Timer Function

23.10.4 Combination of TAUDTTINm Input Period Count Detecting Function and Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

When a capture trigger is input to TAUDTTINm of both channels simultaneously, an overflow from $FFFF_H$ in TAUDnCNTm of the TAUDTTINm input period count detecting function is detected by INTTAUDnIm from the overflow interrupt output function (during TAUDTTINm input period count detection).

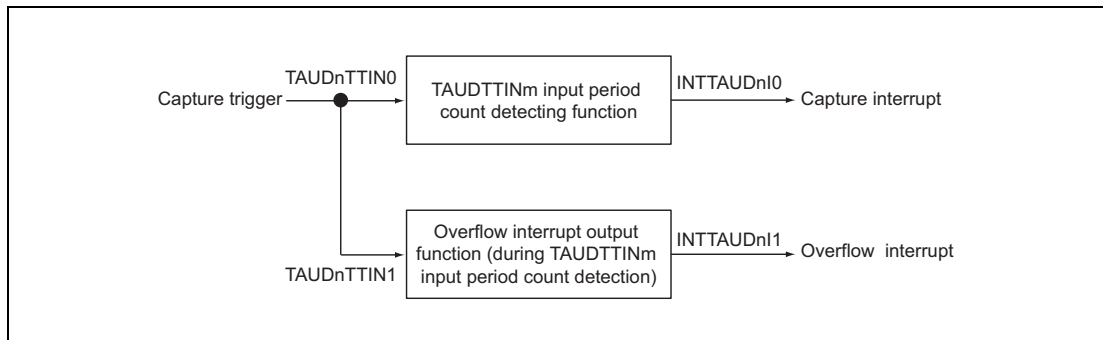


Figure 23.24 Combination of TAUDTTINm Input Period Count Detecting Function and Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Timing diagram

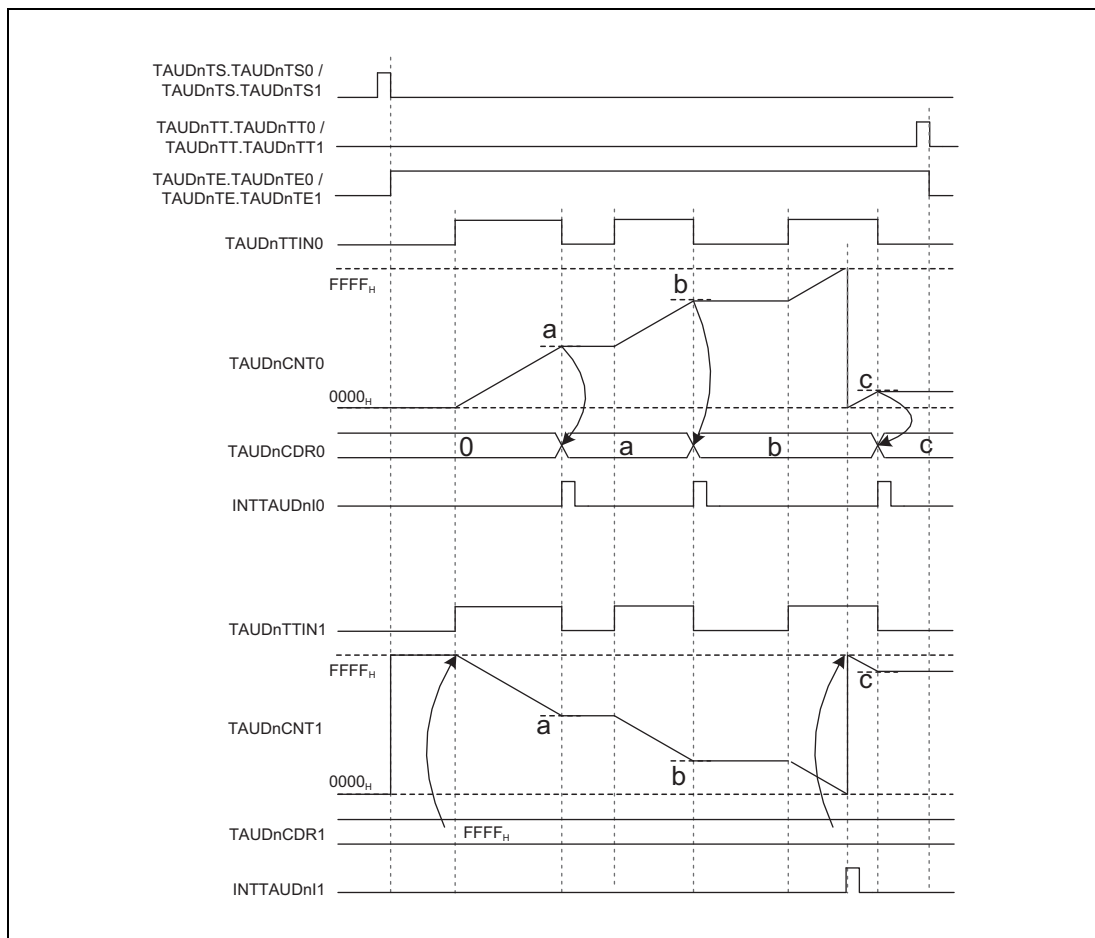


Figure 23.25 Interrupt Generation via Combination of TAUDTTINm Input Period Count Detecting Function and Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

23.11 TAUDnTTINm Edge Detection

Edge detection is based on the operating clock. This means that an edge can only be detected at the next rising edge of the operating clock. This can lead to a maximum delay of one operating clock cycle.

Figure 23.26 shows when edge detection takes place.

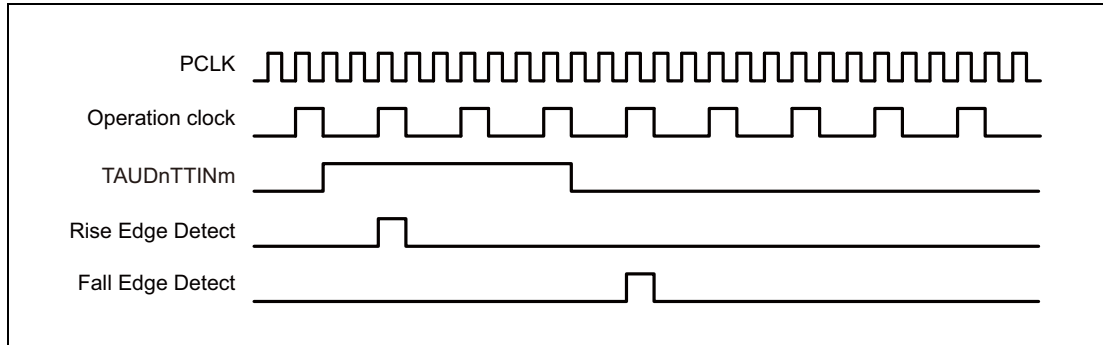


Figure 23.26 Basic Edge Detection Timing

Figure 23.26 shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUDnIm pin and TAUDn causes a delay time.

23.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the Timer Array Unit D. For a general overview of independent channel operation, see **Section 23.2, Overview**.

This section describes functions that generate interrupts at regular intervals or with a specified delay.

23.12.1 Interval Timer Function

23.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals. When an interrupt is generated, the TAUDnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode, refer to **Table 23.44, Contents of the TAUDnCMORm register for Interval Timer Function**
- The channel output mode must be set to Independent Channel Output Mode 1, refer to **Section 23.7, Channel Output Modes**

Functional description

The counter is started by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is written to TAUDnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUDnIm is generated and the TAUDnTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and subsequently continues operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1, which in turn sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDnTTOUTm stop but retain their values. The counter can be restarted by setting TAUDnTS.TAUDnTSM to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSM to 1 during operation.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDnTTOUTm does not toggle. This results in an inverted TAUDnTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details refer to **Section 23.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

23.12.1.2 Equations

$$\text{INTTAUDnIm cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1)$$

$$\text{TAUDnTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1) \times 2$$

23.12.1.3 Block Diagram and General Timing Diagram

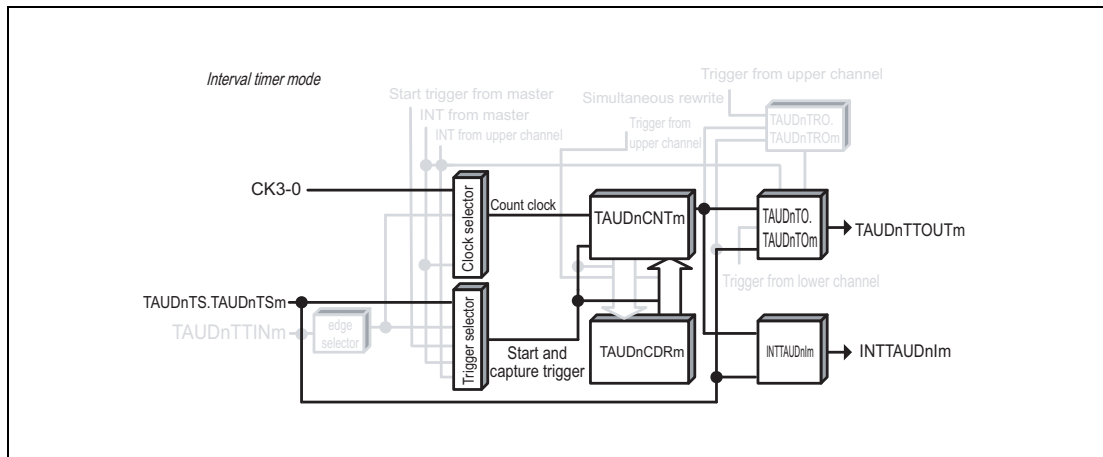


Figure 23.27 Block Diagram of Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1)

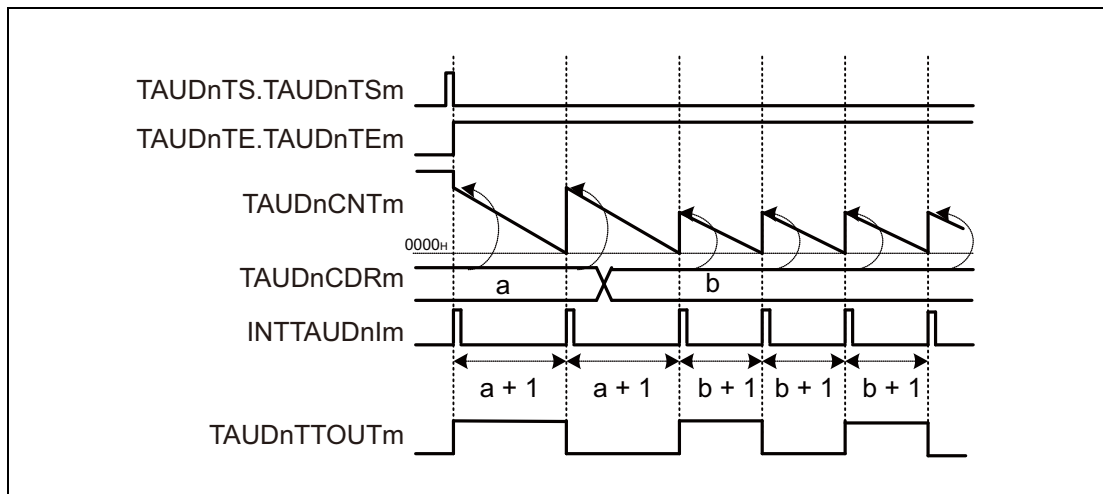


Figure 23.28 General Timing Diagram of Interval Timer Function

23.12.1.4 Register Settings

(1) TAUDnCMORm

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------|-----|----------------|-----|----------|---------------|-----|-----|----------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKS [1:0] | | TAUDnCCS [1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS [1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.44 Contents of the TAUDnCMORm register for Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 15, 14 | TAUDnCKS [1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 000: Triggers the counter by software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated to toggle TAUDnTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDnTTOUTm at the beginning of an operation. |

(2) TAUDnCMURm

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.45 Contents of the TAUDnCMURm register for Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode**Table 23.46 Control Bit Settings in Independent Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: The setting is disabled in toggle mode. (the value after reset). |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDnTOUTm can then be controlled independently of the interrupts. For details, see **Section 23.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the interval timer function. Therefore, these registers should be set to 0.

Table 23.47 Simultaneous Rewrite Settings for Interval Timer Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.1.5 Operating Procedure for Interval Timer Function

Table 23.48 Operating Procedure for Interval Timer Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.44, Contents of the TAUDnCMORm register for Interval Timer Function , and Table 23.45, Contents of the TAUDnCMURm register for Interval Timer Function . | Channel operation is stopped. |
| | Set the value of TAUDnCDRm register. | |
| | Set channel output mode by setting the control bits as described in Table 23.46, Control Bit Settings in Independent Channel Output Mode 1 . | |
| Start Operation | Set TAUDnTS.TAUDnTSM to 1. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDnTTOUTm toggles. |
| | During Operation | <p>The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.</p> <p>TAUDnCNTm counts down. When the counter reaches 0000_H:</p> <ul style="list-style-type: none"> The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. INTTAUDnIm is generated and TAUDnTTOUTm toggles. |
| Stop Operation | Set TAUDnTT.TAUDnTTM to 1. TAUDnTT.TAUDnTTM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

Restart

23.12.1.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H, count clock = PCLK/2

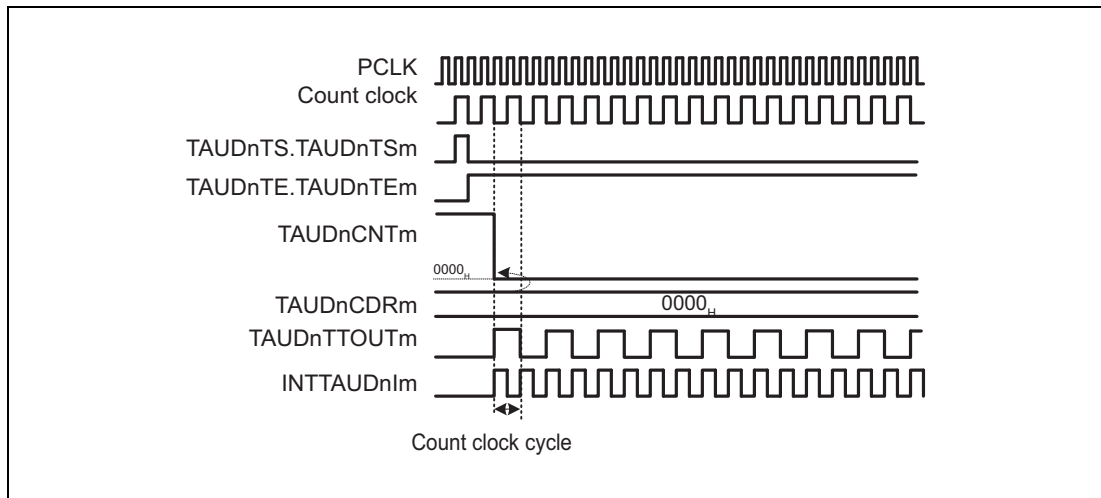


Figure 23.29 TAUDnCDRm = 0000_H, Count Clock = PCLK/2

- If TAUDnCDRm = 0000_H and the count clock = PCLK/2, the TAUDnCDRm value is loaded into TAUDnCNTm every count clock, meaning that TAUDnCNTm is always 0000_H.
- INTTAUDnIm is generated every count clock, resulting in TAUDnTTOUTm toggling every count clock.

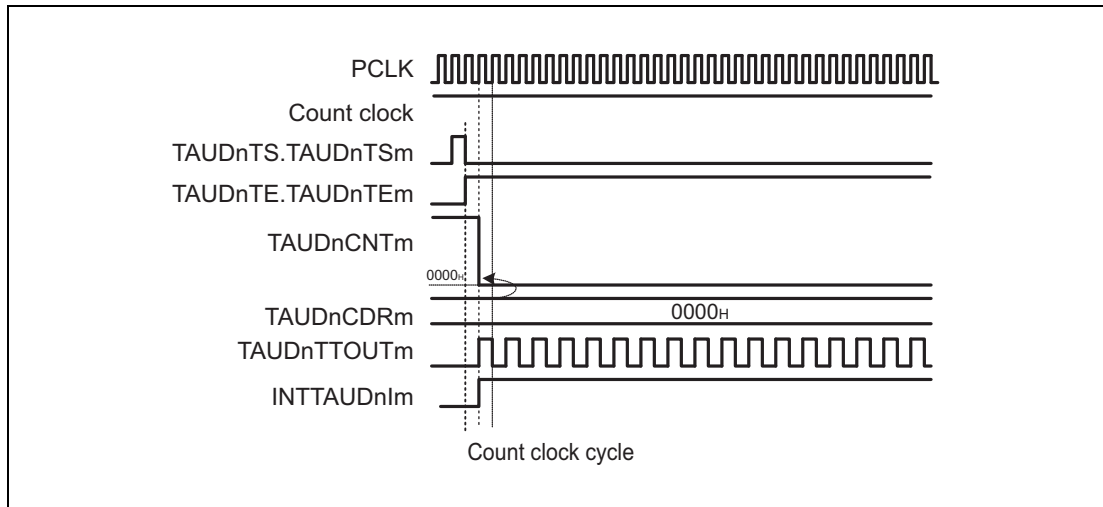
(2) TAUDnCDRm = 0000_H, count clock = PCLK

Figure 23.30 TAUDnCDRm = 0000_H, Count Clock = PCLK

- If TAUDnCDRm = 0000_H and the count clock = PCLK, the TAUDnCDRm value is loaded into TAUDnCNTm every PCLK clock, meaning that TAUDnCNTm is always 0000_H.
- Though the first interrupt is generated, subsequent interrupts are not generated. TAUDTTOUTm is toggled every PCLK clock.

NOTE

INTTAUDnIm is fixed to the high level.

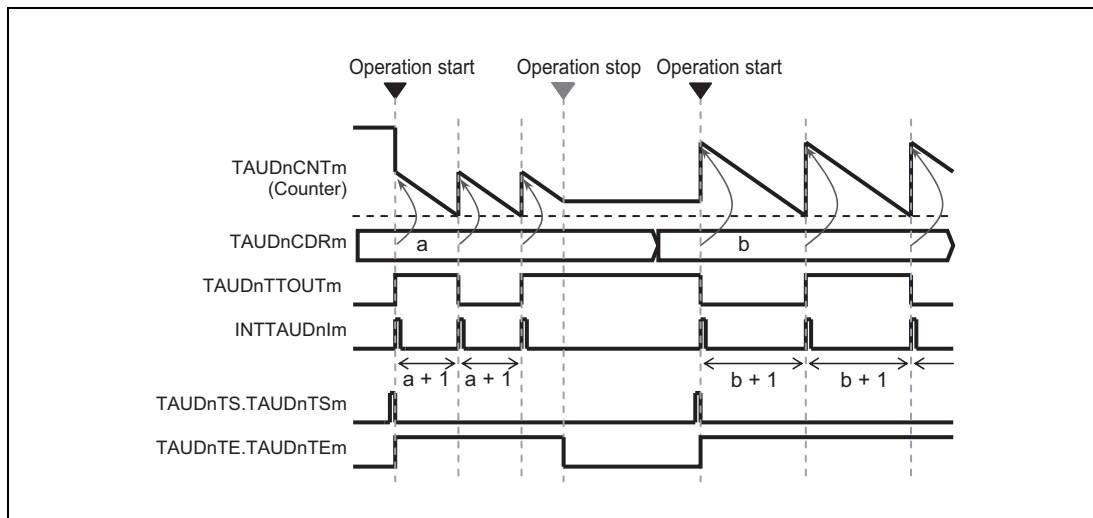
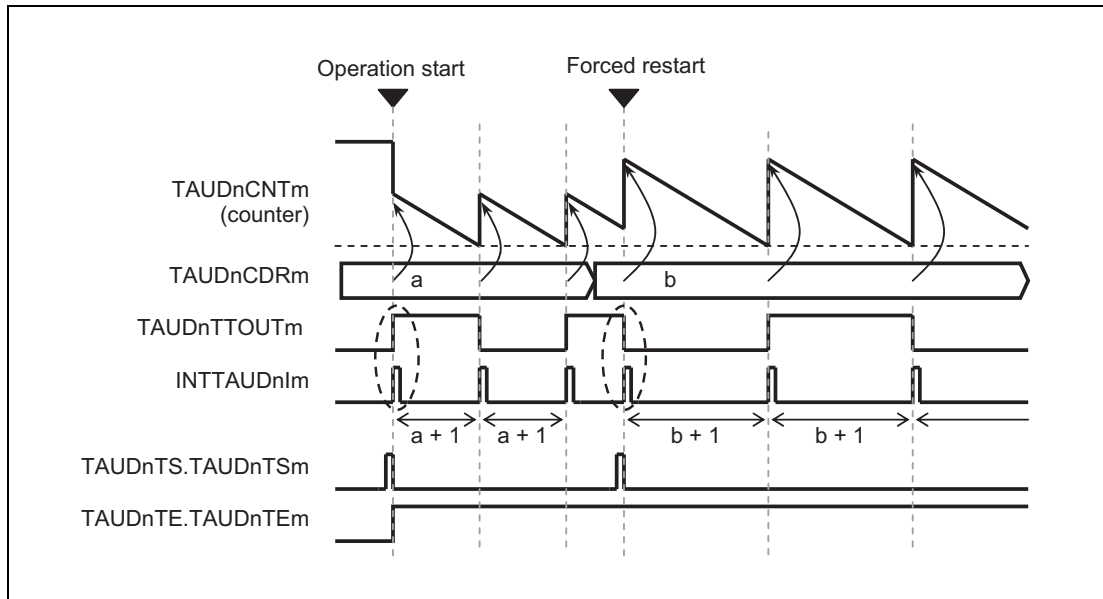
(3) Operation stop and restart

Figure 23.31 Operation Stop and Restart (TAUDnCMORM.TAUDnMD0 = 1)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

(4) Forced restart (TAUDnCMORm.TAUDnMD0 = 1)**Figure 23.32 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1)**

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When the counter is forcibly restarted, the TAUDnCDRm value is reflected in TAUDnCNTm and the counter starts.
A forced restart causes immediate reflection of changes to the value of TAUDnCDRm.
- When the counter is forcibly restarted, an interrupt (INTTAUDnIm) is generated and TAUDnTOUTm is inverted.

(5) Forced restart ($TAUDnCMORm.TAUDnMD0 = 0$)

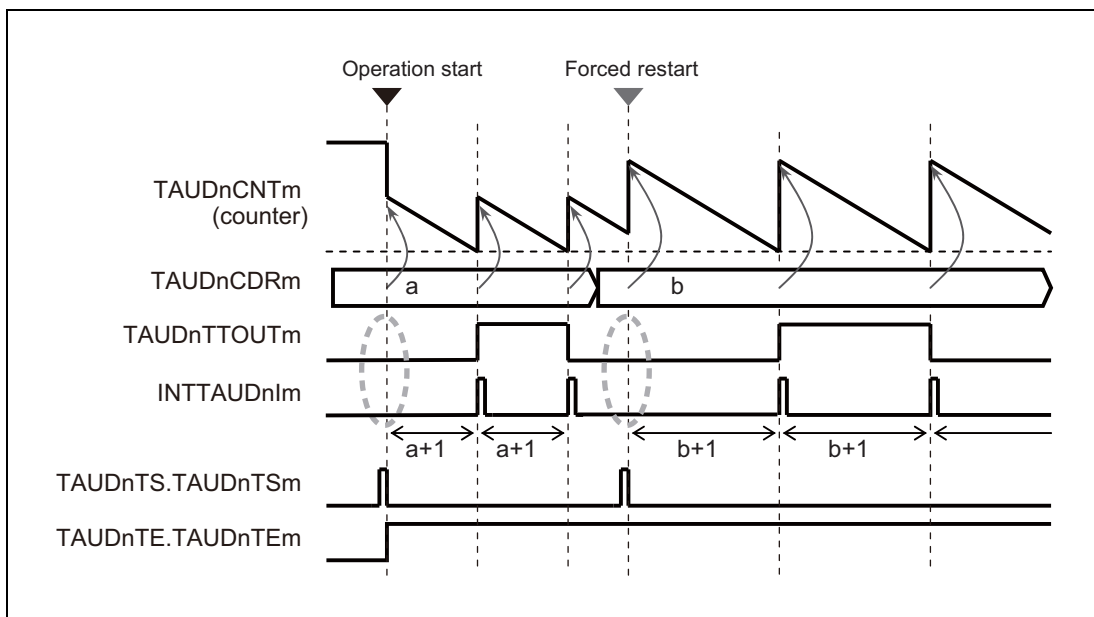


Figure 23.33 Forced Restart Operation ($TAUDnCMORm.TAUDnMD0 = 0$)

- When the counter is forcibly restarted, an interrupt (INTTAUDnIm) is not generated or TAUDnTOUTm is not inverted.

23.12.2 TAUDnTTINm Input Interval Timer Function

23.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals or when a valid TAUDnTTINm input edge is detected. When an interrupt is generated, the TAUDnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operating mode should be set to interval timer mode. See **Table 23.49, Contents of the TAUDnCMORm register for TAUDnTTINm Input Interval Timer Function.**
- The channel output mode should be set to independent channel output mode 1. See **Section 23.7, Channel Output Modes.**

Functional description

This function operates in an identical manner to the interval timer function (see **Section 23.12.1, Interval Timer Function**) except that this function is restarted by a valid TAUDnTTINm input edge. The type of edge used as a trigger is specified using the TAUDnCMURm.TAUDnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edge can be selected.

23.12.2.2 Equations

$$\text{INTTAUDnIm cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1)$$

$$\text{TAUDnTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1) \times 2$$

23.12.2.3 Block Diagram and General Timing Diagram

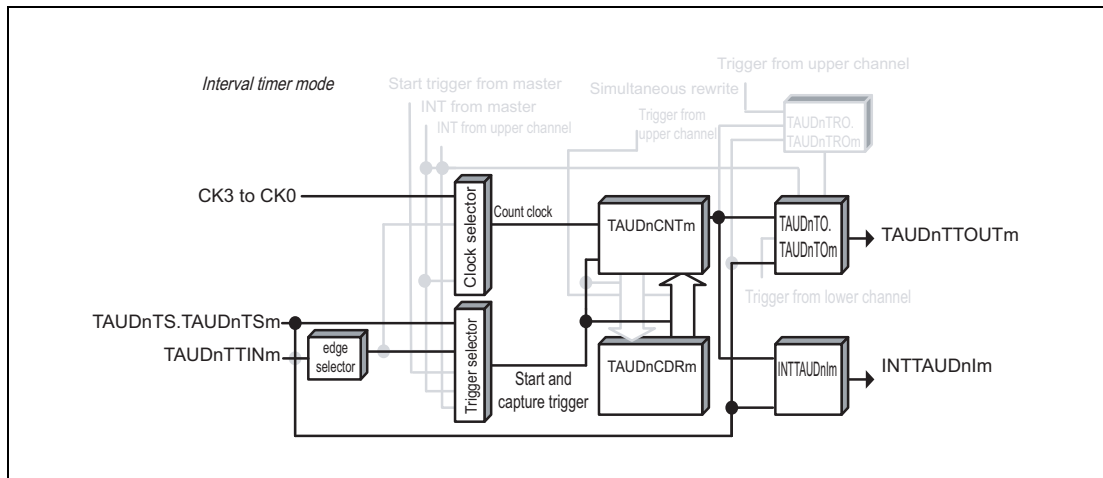


Figure 23.34 Block Diagram of TAUDnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).
- Rising edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

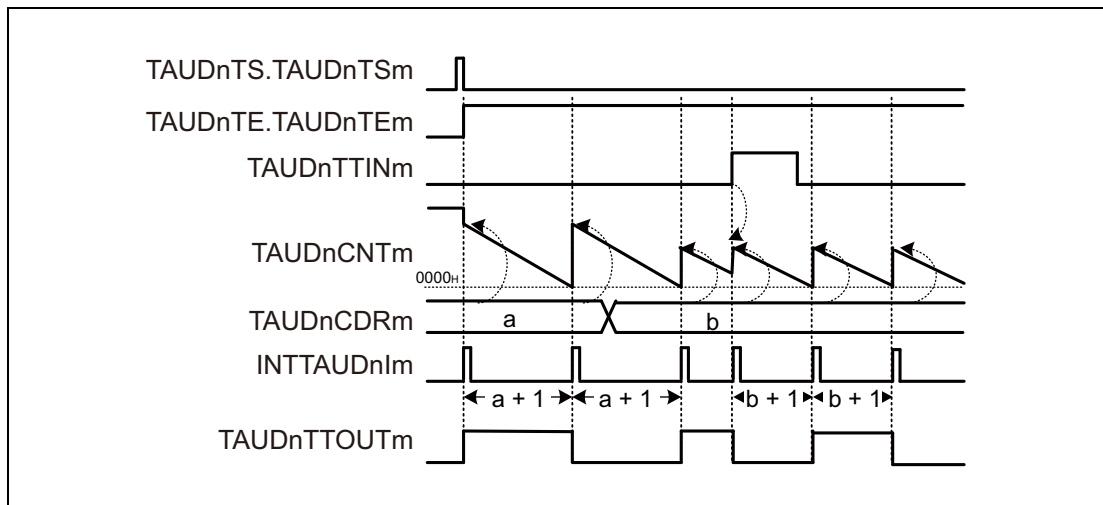


Figure 23.35 General Timing Diagram of TAUDnTTINm Input Interval Timer Function

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|----------------|-----|----------------|-----|----------|---------------|-----|-----|----------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS [1:0] | | TAUDnCCS [1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS [1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.49 Contents of the TAUDnCMORm register for TAUDnTTINm Input Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 15, 14 | TAUDnCKS [1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid TAUDnTTINm input edge signal is used as an external start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated to toggle TAUDnTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDnTTOUTm at the beginning of an operation. |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.50 Contents of the TAUDnCMURm register for TAUDnTTINm Input Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges |

(3) Channel output mode**Table 23.51 Control Bit Settings in Independent Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: The setting is disabled in toggle mode. (the value after reset). |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled |
| TAUDnTDL.TAUDnTDLm | (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled |
| TAUDnTRC.TAUDnTRCm | (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDnTTOUTm can then be controlled independently of the interrupts. For details, see **Section 23.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm Input Interval Timer Function. Therefore, these registers should be set to 0.

Table 23.52 Simultaneous Rewrite Settings for TAUDnTTINm Input Interval Timer Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled |
| TAUDnRDM.TAUDnRDMm | (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDC.TAUDnRDCm | |

23.12.2.4 Operating Procedure for TAUDnTTINm Input Interval Timer Function

Table 23.53 Operating Procedure for TAUDnTTINm Input Interval Timer Function

| | Operation | TAUDn Status |
|-------------------------|---|--|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.49, Contents of the TAUDnCMORm register for TAUDnTTINm Input Interval Timer Function, and Table 23.50, Contents of the TAUDnCMURm register for TAUDnTTINm Input Interval Timer Function. | Channel operation is stopped. |
| | Set the value of TAUDnCDRm register. | |
| | Set channel output mode by setting the control bits as described in Table 23.51, Control Bit Settings in Independent Channel Output Mode 1. | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSM to 1. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDnTTOUTm toggles. |
| | During Operation The values of the TAUDnCMURm.TAUDnTIS[1:0] and the TAUDnCDRm register are changeable at any time. The TAUDnCNTm register can be read at all times. Detection of TAUDnTTINm edge | TAUDnCNTm counts down. When the counter reaches 0000H: <ul style="list-style-type: none"> The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. INTTAUDnIm is generated and TAUDnTTOUTm toggles. When a TAUDnTTINm input valid edge is detected during count operation, the TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTM to 1. TAUDnTT.TAUDnTTM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.12.2.5 Specific Timing Diagrams

The timing diagrams in **Section 23.12.1, Interval Timer Function** apply, and in addition, the counter can also be restarted by a valid TAUDnTTINm input edge without using this function.

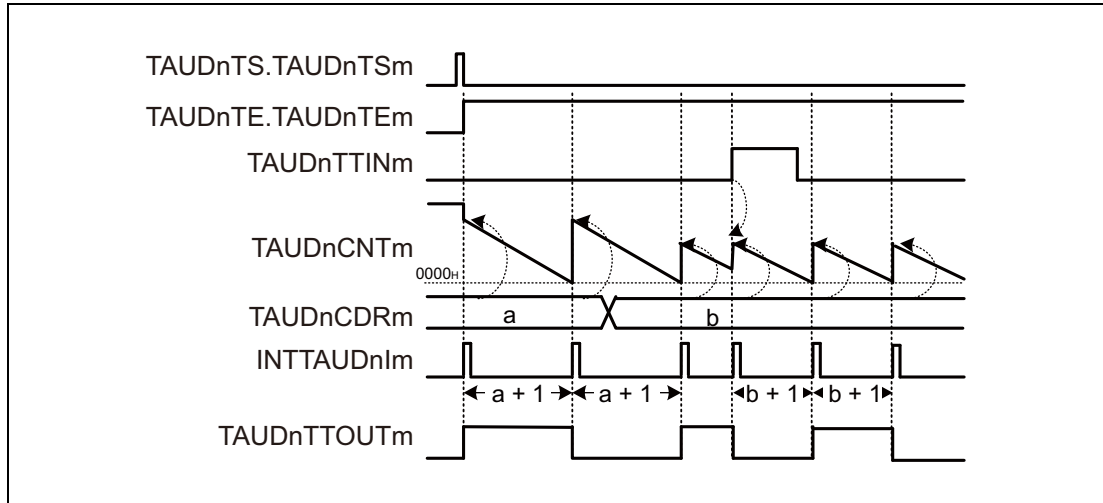


Figure 23.36 Counter Triggered by Rising TAUDnTTINm Input Edge
(TAUDnCMURm.TAUDnTIS[1:0] = 01_B), TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDnTTINm input edge is detected, an interrupt is generated which causes TAUDnTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B).

23.12.3 Clock Divider Function

23.12.3.1 Overview

Summary

This function is used as a frequency divider. The frequency of the input signal TAUDnTTINm is divided by a factor related to TAUDnCDRm, and the resulting signal is output to TAUDnTTOUTm.

Prerequisites

- TAUDnTTINm should have a fixed frequency.
- The operating mode should be set to interval timer mode. (See **Table 23.54, Contents of the TAUDnCMORm register for Clock Divider Function.**)
- The channel output mode should be set to independent channel output mode 1. (See **Section 23.7, Channel Output Modes.**)

Functional description

The counter is started by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value, using TAUDnTTINm as a count clock.

When the counter value reaches 0000_H, INTTAUDnIm occurs and TAUDnTTOUTm signal is toggled. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The value of TAUDnCDRm can be rewritten at any time. The changed value of TAUDnCDRm is applied when the counter starts to count down next time.

The counter can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEM = 0. TAUDnCNTm and TAUDnTTOUTm stop but retain their values. The function can be restarted by setting TAUDnTS.TAUDnTSM = 1. The counter can also be forcibly restarted without making a stop by setting TAUDnTS.TAUDnTSM = 1 during operation (forced restart).

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDnTTOUTm does not toggle. This results in an inverted TAUDnTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details, see **Section 23.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

NOTE

TAUDnTTINm input signals are sampled at the frequency of the operating clock set by TAUDnCMORm.TAUDnCKS[1:0] bits. Therefore, the TAUDnTTOUTm output clock cycle has an error of ±1 operating clock cycle.

23.12.3.2 Equations

- When rising edge detection is selected:
 $TAUDnTTOUTm \text{ frequency} = TAUDnTTINm \text{ frequency} / [(TAUDnCDRm + 1) \times 2]$
- When falling edge detection is selected:
 $TAUDnTTOUTm \text{ frequency} = TAUDnTTINm \text{ frequency} / [(TAUDnCDRm + 1) \times 2]$
- When falling and rising edge detection is selected:
 $TAUDnTTOUTm \text{ frequency} = TAUDnTTINm \text{ frequency} / (TAUDnCDRm + 1)$

23.12.3.3 Block Diagram and General Timing Diagram

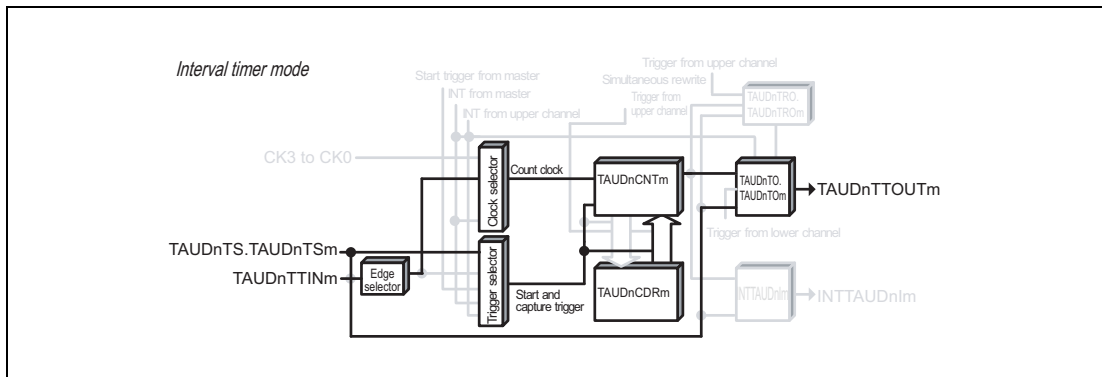


Figure 23.37 Block Diagram of Clock Divider Function

The following settings apply to the general timing diagram.

- INTTAUDnIm generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

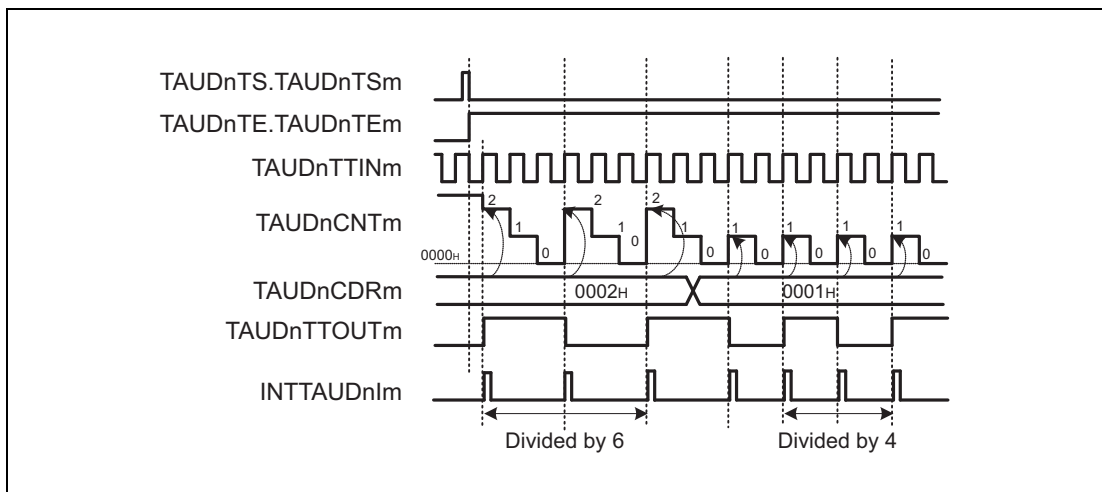


Figure 23.38 General Timing Diagram of Clock Divider Function

23.12.3.4 Register Settings

(1) TAUDnCMORm

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------------------|-----|-------------------|-----|--------------|---------------|-----|-----|-------------------|-----|---|--------------|-----|-----|-----|--------------|
| | TAUDnCKS [1:0] | | TAUDnCCS [1:0] | | TAUDn MAS | TAUDnSTS[2:0] | | | TAUDnCOS [1:0] | | — | TAUDnMD[4:1] | | | | TAUDn MD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.54 Contents of the TAUDnCMORm register for Clock Divider Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 01: Valid TAUDnTTINm input edge is used as a count clock. |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated to toggle TAUDnTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDnTTOUTm is toggled at the beginning of operation. |

(2) TAUDnCMURm

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.55 Contents of the TAUDnCMURm register for Clock Divider Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges |

(3) Channel output mode**Table 23.56 Control Bit Settings in Independent Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: The setting is disabled in toggle mode. (the value after reset). |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 23.57 Simultaneous Rewrite Settings for Clock Divider Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.3.5 Operating Procedure for Clock Divider Function

Table 23.58 Operating Procedure for Clock Divider Function

| | Operation | TAUDn Status |
|-------------------------|---|--|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.54, Contents of the TAUDnCMORm register for Clock Divider Function , and Table 23.55, Contents of the TAUDnCMURm register for Clock Divider Function . | Channel operation is stopped. |
| | Set the value of TAUDnCDRm register. | |
| | Set channel output mode by setting the control bits as described in Table 23.56, Control Bit Settings in Independent Channel Output Mode 1 . | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 is set to 1, INTTAUDnIm occurs and TAUDnTTOUTm is toggled. |
| | During Operation The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times. | TAUDnCNTm counts down each time TAUDnTTINm input edge is detected. When the counter reaches 0000 _H : <ul style="list-style-type: none"> • TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. • INTTAUDnIm is generated. • TAUDnTTOUTm is toggled. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDnTTOUTm retain their current values. |

23.12.3.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000H

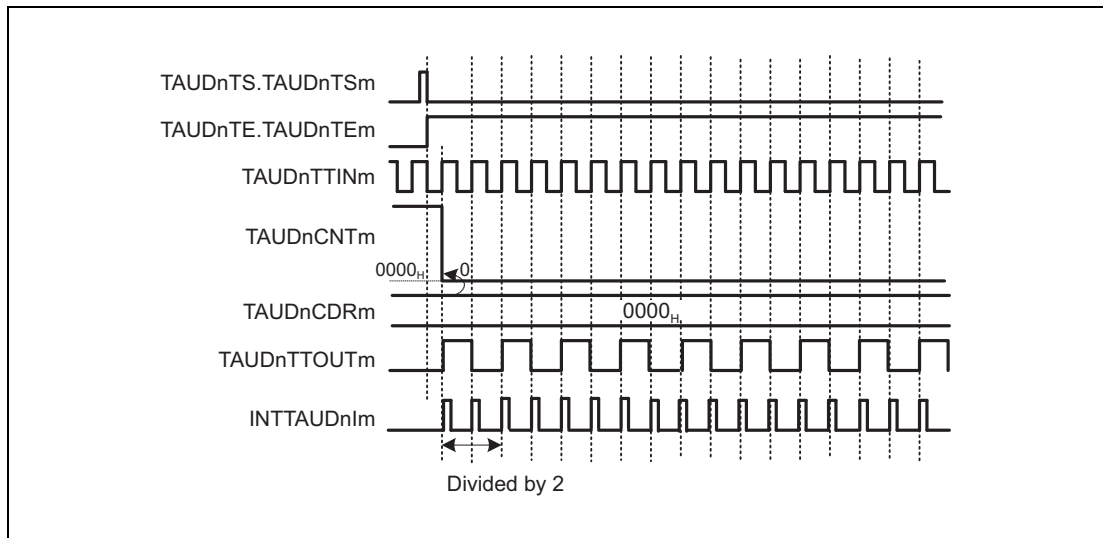


Figure 23.39 TAUDnCDRm = 0000H, TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01B

- If TAUDnCDRm is 0000H, TAUDnCNTm is always 0000H.
- INTTAUDnIm is generated every count clock, resulting in TAUDnTTOUTm toggling every count clock

Figure 23.39 shows an operation timing example. Actually, there is a delay from TINm detection until TOUTm output because of the delay time of a noise filter or synchronization circuit placed between the TAUDnIm pin and TAUDn.

(2) Operation restart

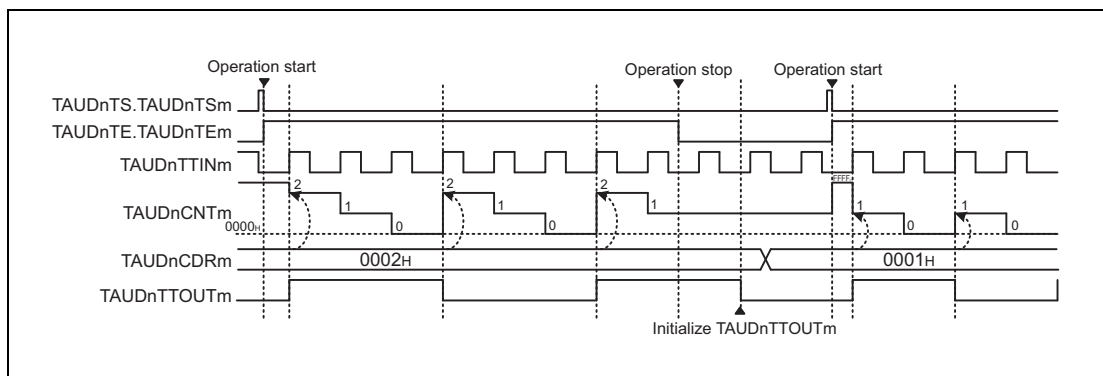


Figure 23.40 Operation Restart (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01B)

To reset the value of TAUDnTTOUTm:

- Set TAUDnTOE.TAUDnTOEm = 0 when the counter is stopped (TAUDnTE.TAUDnTEm = 0).
- Then, write either 0 or 1 to TAUDnTO.TAUDnTOM to set the new start value of TAUDnTTOUTm.

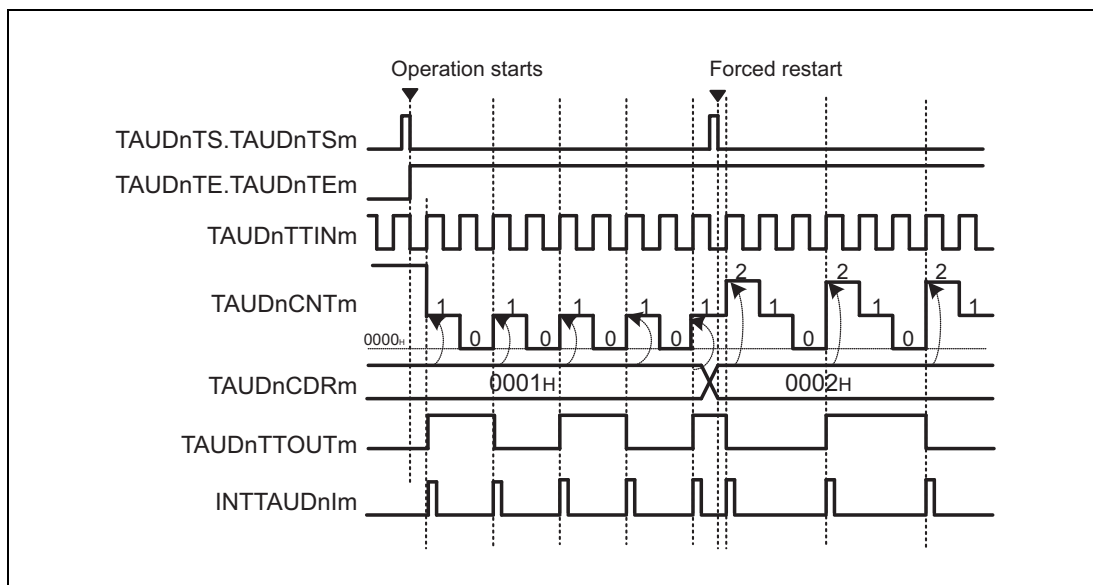
(3) Forced restart

Figure 23.41 Forced Restart Operation
 (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm = 1 during operation.
- The value of TAUDnCDRm is written to TAUDnCNTm and the count operation restarts.
- TAUDnTTOUTm restarts at the same level as before the forced restart.

23.12.4 External Event Count Function

23.12.4.1 Overview

Summary

This function is used as an event timer, which generates an interrupt (INTTAUDnIm) when a specific number of TAUDnTTINm input valid edge has been detected.

Prerequisites

- The operating mode should be set to the event count mode. (See **Table 23.59, Contents of the TAUDnCMORm register for External Event Count Function.**)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When the counter starts, the current value of TAUDnCDRm is loaded into TAUDnCNTm.

When a valid TAUDnTTINm input edge is detected, the value of TAUDnCNTm decrements by 1. TAUDnCNTm retains this value until a valid TAUDnTTINm input edge is detected or the counter is restarted.

When the valid edge is detected for the (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUDnTS.TAUDnTSm to 1 during operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

An edge type used as a trigger is specified by TAUDnCMURm.TAUDnTIS[1:0] bits.

- When TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 10_B, both edges are counted.

23.12.4.2 Equations

Number of valid edges detected before INTTAUDnIm generation = TAUDnCDRm + 1

23.12.4.3 Block Diagram and General Timing Diagram

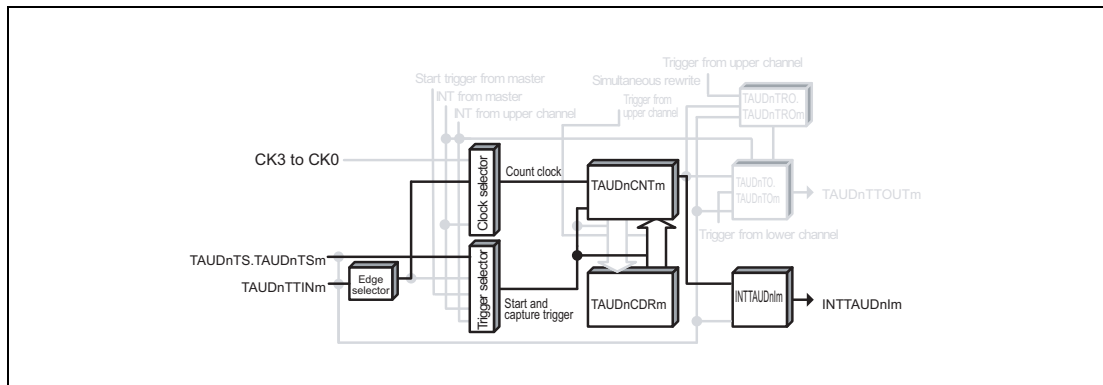


Figure 23.42 Block Diagram of External Event Count Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

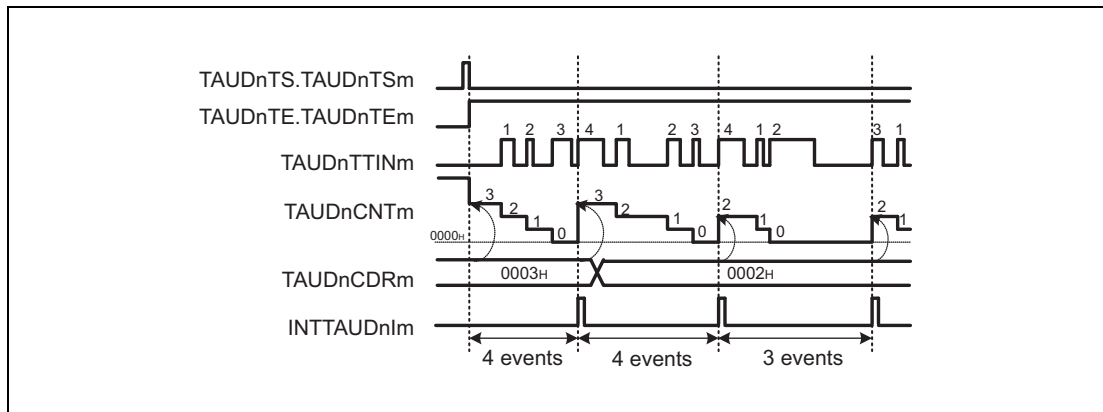


Figure 23.43 General Timing Diagram of External Event Count Function

23.12.4.4 Register Settings

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------|-----|-------------------|-----|--------------|---------------|-----|-----|-------------------|-----|---|--------------|-----|-----|-----|--------------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS [1:0] | | TAUDnCCS [1:0] | | TAUDn MAS | TAUDnSTS[2:0] | | | TAUDnCOS [1:0] | | — | TAUDnMD[4:1] | | | | TAUDn MD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.59 Contents of the TAUDnCMORm register for External Event Count Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 01: Valid TAUDnTTINm input edge is used as a count clock. |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0011: Event count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.60 Contents of the TAUDnCMURm register for External Event Count Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of falling and rising edges |

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the this function. Therefore, these registers should be set to 0.

Table 23.61 Simultaneous Rewrite Settings for External Event Count Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.4.5 Operating Procedure for External Event Count Function

Table 23.62 Operating Procedure for External Event Count Function

| Operation | TAUDn Status |
|--|--|
| Initial Channel Setting Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.59, Contents of the TAUDnCMORm register for External Event Count Function , and Table 23.60, Contents of the TAUDnCMURm register for External Event Count Function . Set the value of TAUDnCDRm register. | Channel operation is stopped. |
| Restart Start Operation Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value and waits for TAUDnTTINm input edge detection. |
| During Operation Detection of TAUDnTTINm edge The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at any time. | TAUDnCNTm counts down each time TAUDnTTINm input edge is detected. When an effective edge is detected (TAUDnCDRm + 1) times: <ul style="list-style-type: none"> • TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. • INTTAUDnIm is generated. Afterwards, this procedure is repeated. |
| Stop Operation Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value. |

23.12.4.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

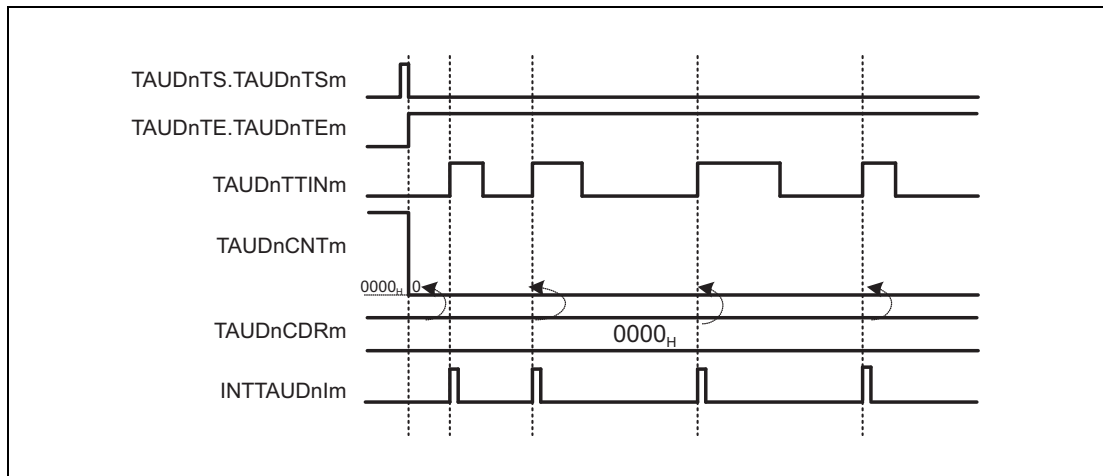


Figure 23.44 TAUDnCDRm = 0000_H, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If 0000_H = TAUDnCDRm, 0000_H is loaded into TAUDnCNTm each time a valid TAUDnTTINm input edge is detected.
In other words, INTTAUDnIm occurs each time a valid TAUDnTTINm input edge is detected.

(2) Operation stop and restart

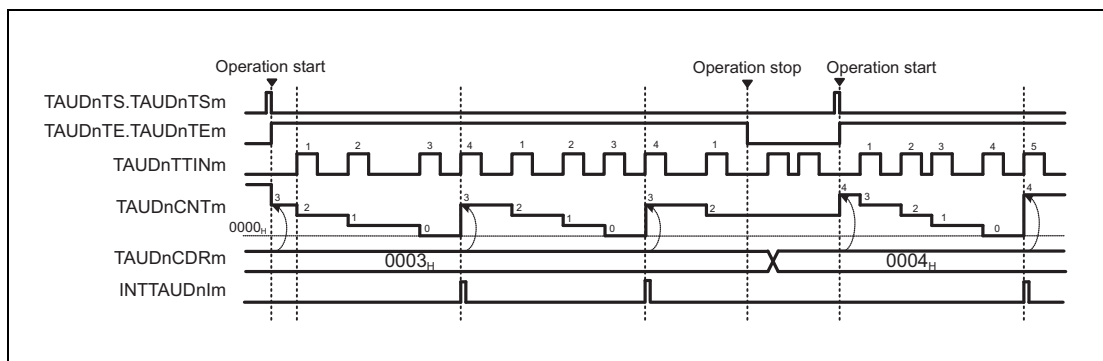


Figure 23.45 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEM to 0.
- TAUDnCNTm stops and retains its current value. TAUDnTTINm continues and TAUDnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUDnTS.TAUDnTSM to 1. TAUDnCNTm loads the TAUDnCDRm value and restarts count operation.

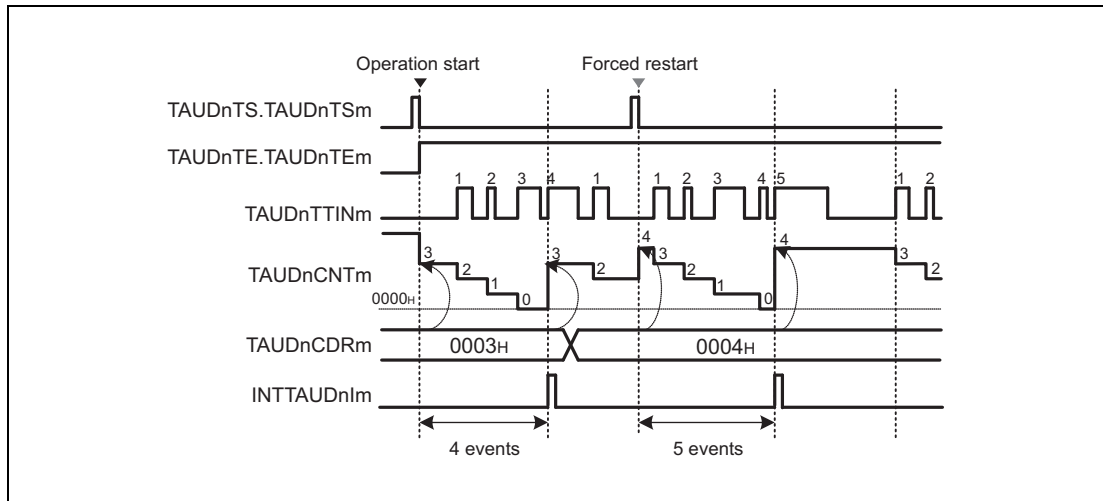
(3) Forced restart

Figure 23.46 Forced Restart Operation (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

Once a forced restart is made, the changed TAUDnCDRm value is applied to TAUDnCNTm.

- The counter can be restarted without making a stop by setting TAUDnTS.TAUDnTSm to 1 during operation.
- The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter awaits the next valid TAUDnTTINm input edge.

23.12.5 Delay Count Function

23.12.5.1 Overview

Summary

This function generates interrupts (INTTAUDnIm), which have a defined delay to the TAUDnTTINm input signal. TAUDnTTINm input signal pulses that occur within the delay period are ignored.

Prerequisites

- The operating mode should be set to one-count mode. See **Table 23.63, Contents of the TAUDnCMORm register for Delay Count Function.**
- TAUDnTTOUTm is not used with this function.
- Start trigger detection should be disabled during counting (TAUDnCMORm.TAUDnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt is generated. The counter returns to FFFF_H and awaits the next valid TAUDnTTINm input edge.

When the counter is counting down, further TAUDnTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits:

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

23.12.5.2 Equations

Delay between TAUDnTTINm and INTTAUDnIm = count clock cycle × (TAUDnCDRm + 1)

23.12.5.3 Block Diagram and General Timing Diagram

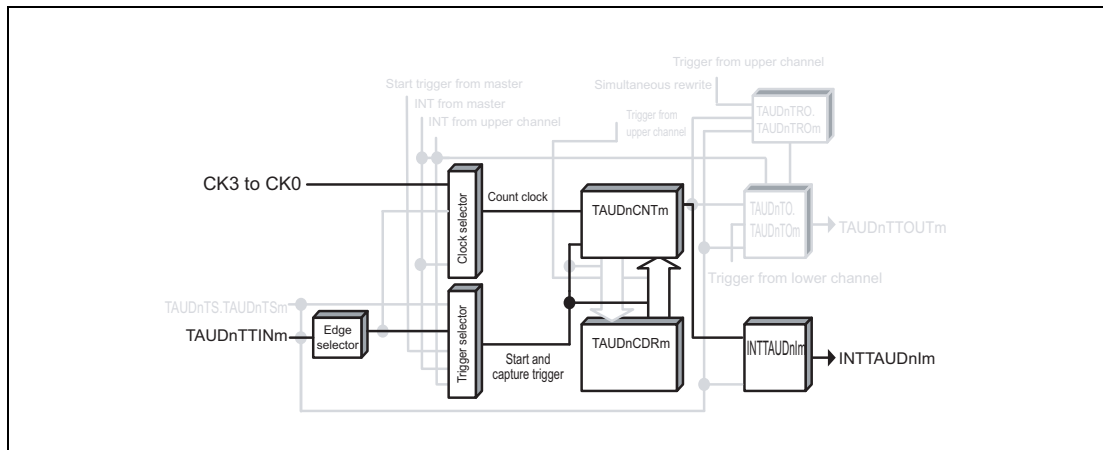


Figure 23.47 Block Diagram of Delay Count Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

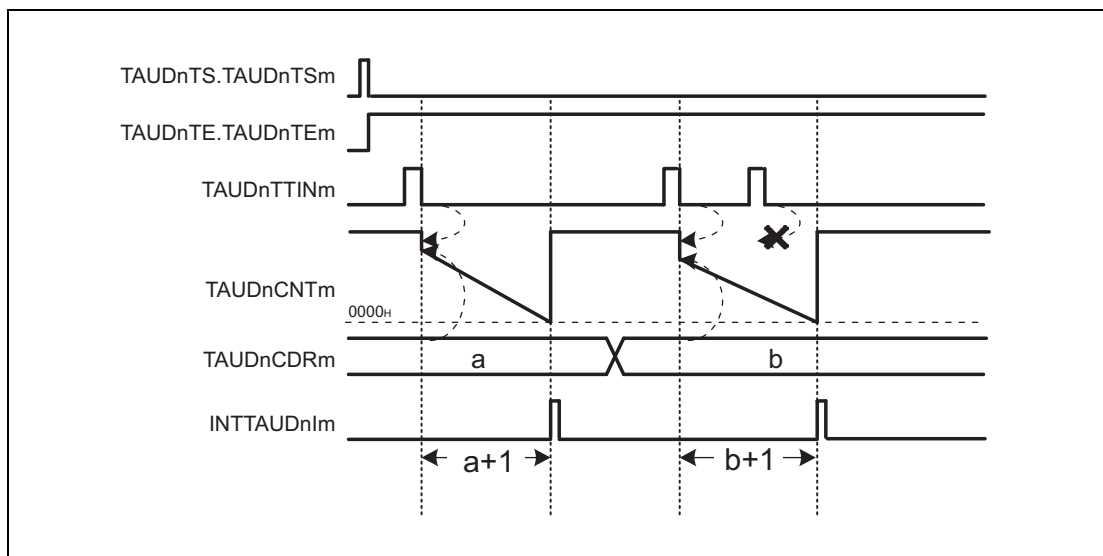


Figure 23.48 General Timing Diagram of Delay Count Function

23.12.5.4 Register Settings

(1) TAUDnCMORm

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.63 Contents of the TAUDnCMORm register for Delay Count Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid TAUDnTTINm input edge signal is used as an external start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 0: Disables a start trigger during operation. |

(2) TAUDnCMURm

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.64 Contents of the TAUDnCMURm register for Delay Count Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 23.65 Simultaneous Rewrite Settings for Delay Count Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.5.5 Operating Procedure for Delay Count Function**Table 23.66 Operating Procedure for Delay Count Function**

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.63, Contents of the TAUDnCMORm register for Delay Count Function , and Table 23.64, Contents of the TAUDnCMURm register for Delay Count Function . | Channel operation is stopped. |
| | Set the value of TAUDnCDRm register. | |
| Start Operation | Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. |
| | Detection of TAUDnTTINm start edge | When a start edge is detected, the TAUDnCDRm value is loaded in TAUDnCNTm. |
| During Operation | The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times. | TAUDnCNTm counts down. When the counter reaches 0000 _H , INTTAUDnIm is generated. TAUDnCNTm stops counting, returns FFFF _H , and waits for a trigger. |
| | | If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its value. |

Restart

23.12.6 One-Pulse Output Function

23.12.6.1 Overview

Summary

This function generates an interrupt (INTTAUDnIm) when a valid TAUDnTTINm input edge is detected and at a defined interval afterward. TAUDnTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUDnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operating mode should be set to pulse one-count mode. (See **Table 23.67, Contents of the TAUDnCMORm register for One-Pulse Output Function.**)
- The channel output mode should be set to independent channel output mode 2. (See **Section 23.7, Channel Output Modes.**)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. An interrupt is generated and TAUDnTTOUTm is set to the active level.

When the counter reaches 0001_H, an interrupt is generated and TAUDnTTOUTm is set to the inactive level. The counter stops at 0000_H and awaits the next valid TAUDnTTINm input edge.

When the counter is counting down, further TAUDnTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits:

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

23.12.6.2 Equations

Interval between TAUDnTTINm and INTTAUDnIm = TAUDnTTOUTm (timer output) width = count clock cycle × TAUDnCDRm

23.12.6.3 Block Diagram and General Timing Diagram

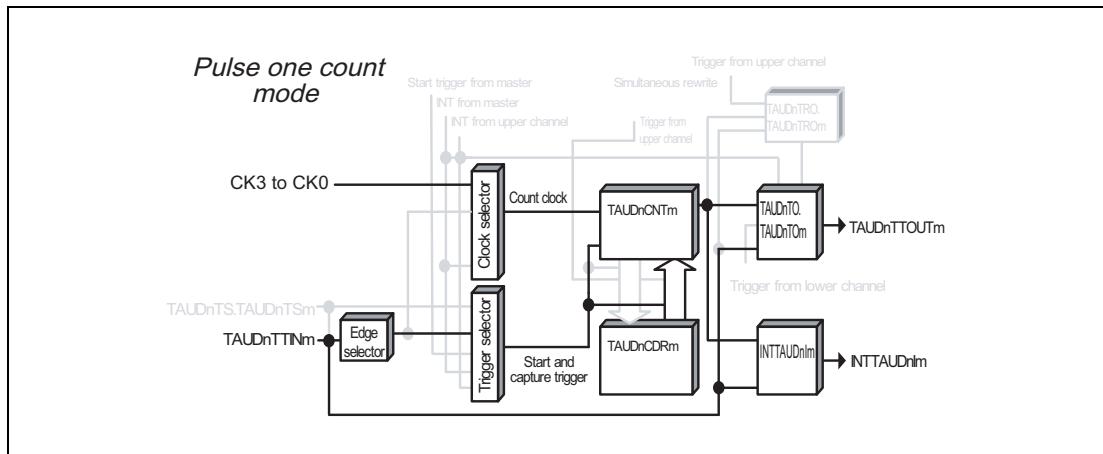


Figure 23.49 Block Diagram of One-Pulse Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

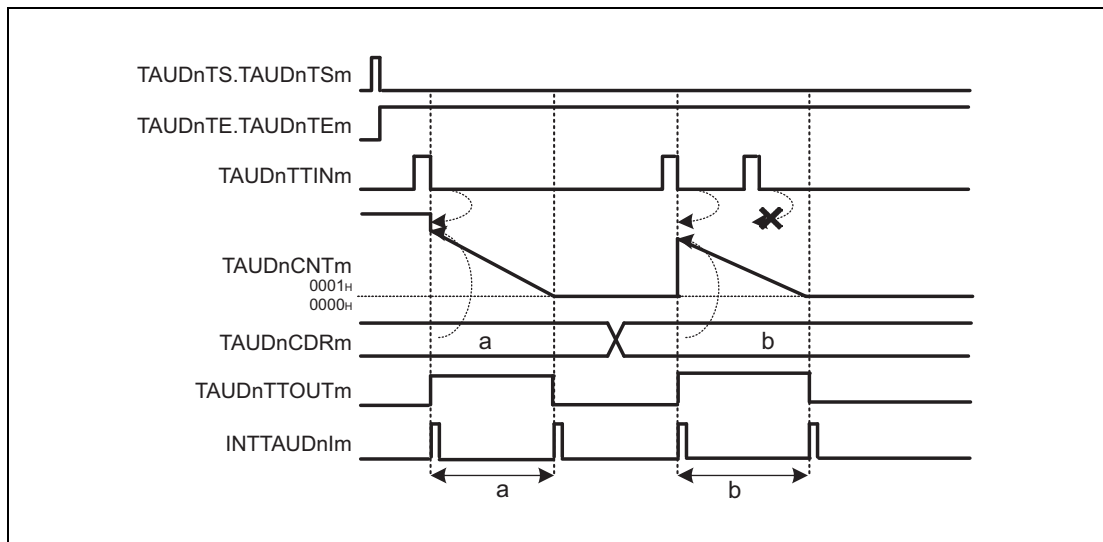


Figure 23.50 General Timing Diagram of One-Pulse Output Function

23.12.6.4 Register Settings

(1) TAUDnCMORm

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.67 Contents of the TAUDnCMORm register for One-Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid TAUDnTTINm input edge signal is used as an external start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1010: Pulse one-count mode |
| 0 | TAUDnMD0 | 0: Disables a start trigger during operation. |

(2) TAUDnCMURm

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.68 Contents of the TAUDnCMURm register for One-Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited |

(3) Channel output mode**Table 23.69 Control Bit Settings in Independent Channel Output Mode 2**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode controlled by software. |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDnTTOUTm can then be controlled independently of the interrupts. For details, see **Table 23.43, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the One-Pulse Output Function. Therefore, these registers should be set to 0.

Table 23.70 Simultaneous Rewrite Settings for One-Pulse Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.6.5 Operating Procedure for One-Pulse Output Function

Table 23.71 Operating Procedure for One-Pulse Output Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.67, Contents of the TAUDnCMORm register for One-Pulse Output Function, and Table 23.68, Contents of the TAUDnCMURm register for One-Pulse Output Function. | Channel operation is stopped. |
| | Set the value of TAUDnCDRm register. | |
| | Set channel output mode by setting the control bits as described in Table 23.69, Control Bit Settings in Independent Channel Output Mode 2. | |
| Start Operation | Set TAUDnTS.TAUDnTSM to 1. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. |
| | Detection of TAUDnTTINm start edge | When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value. |
| During Operation | The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times. | INTTAUDnIm is generated when TAUDnCNTm starts and TAUDnTTOUTm is set to its active level. TAUDnCNTm counts down. When the counter reaches 0001 _H : <ul style="list-style-type: none"> • INTTAUDnIm is generated. • TAUDnTTOUTm is set to its inactive level. TAUDnCNTm stops counting and waits for a trigger. |
| | | If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored. |
| Stop Operation | Set TAUDnTT.TAUDnTTM to 1. TAUDnTT.TAUDnTTM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |



23.12.7 TAUDnTTINm Input Pulse Interval Measurement Function

23.12.7.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUDnCSRm.TAUDnOVF to measure the interval of the TAUDnTTINm input signals.

Prerequisites

- The operating mode should be set to capture mode. See **Table 23.73, Contents of the TAUDnCMORm register for TAUDnTTINm Input Pulse Interval Measurement Function**.
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is started by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDnTTINm edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches FFFF_H before a valid TAUDnTTINm edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 23.72 Effects of Overflow

| TAUDnCMORm. TAUDnCOS[1:0] | When Overflow Occurs | | When a Valid TAUDnTTINm Input is Detected | |
|------------------------------|--------------------------|------------------------|---|------------------------|
| | TAUDnCDRm | TAUDnCSRm. TAUDnOVF | TAUDnCDRm, TAUDnCNTm | TAUDnCSRm. TAUDnOVF |
| 00 | Unchanged | 0 | TAUDnCNTm loaded into TAUDnCDRm | 1 |
| 01 | | 1 | | |
| 10 | Set to FFFF _H | 0 | TAUDnCNTm set to 0, TAUDnCDRm unchanged | Unchanged |
| 11 | | 1 | | |

When TAUDnCMORm.TAUDnCOS[0] = 1, the overflow bit (TAUDnCSRm.TAUDnOVF) can be cleared only by setting TAUDnCSCm.TAUDnCLOV = 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the interval of the TAUDnTTINm signal. However, if an overflow occurs multiple times before a valid TAUDnTTINm input is detected, the overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

The function can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm stops but retains its value. While the function is stopped, valid TAUDnTTINm input edge detection and TAUDnCNTm capture are not performed.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 23.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

NOTE

When TAUDnCMORm.TAUDnCOS[1] = 1, the value of TAUDnCNTm is not loaded into TAUDnCDRm when the first valid TAUDnTTINm input edge occurs after an overflow. However, an interrupt is generated.

23.12.7.2 Equations

TAUDnTTINm input pulse interval = count clock cycle \times [(TAUDnCSRm.TAUDnOVF \times (FFFF_H + 1)) + TAUDnCDRm capture value + 1]

23.12.7.3 Block Diagram and General Timing Diagram

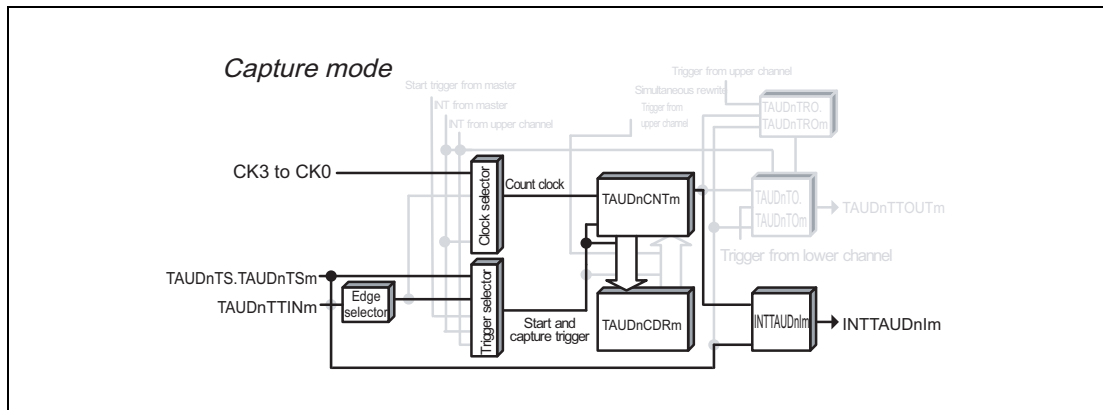


Figure 23.51 Block Diagram of TAUDnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUDnIm not generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 0)
- Falling edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)
- When a valid TAUDnTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1 (TAUDnCMORm.TAUDnCOS[1:0] = 00_B)

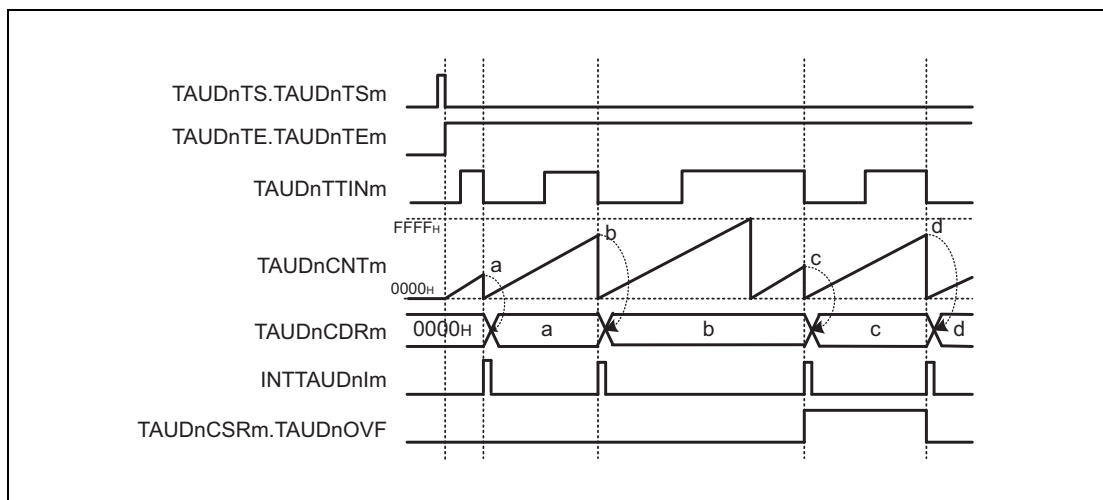


Figure 23.52 General Timing Diagram of TAUDnTTINm Input Pulse Interval Measurement Function

23.12.7.4 Register Settings

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.73 Contents of the TAUDnCMORm register for TAUDnTTINm Input Pulse Interval Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Independent operation. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid edge of the TAUDnTTINm input signal is used as the external capture trigger. |
| 7, 6 | TAUDnCOS[1:0] | See Table 23.72, Effects of Overflow. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0010: Capture mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.74 Contents of the TAUDnCMURm register for TAUDnTTINm Input Pulse Interval Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input pulse interval measurement function. Therefore, these registers should be set to 0.

Table 23.75 Simultaneous Rewrite Settings for TAUDnTTINm Input Pulse Interval Measurement Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.7.5 Operating Procedure for TAUDnTTINm Input Pulse Interval Measurement Function

Table 23.76 Operating Procedure for TAUDnTTINm Input Pulse Interval Measurement Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.73, Contents of the TAUDnCMORm register for TAUDnTTINm Input Pulse Interval Measurement Function , and Table 23.74, Contents of the TAUDnCMURm register for TAUDnTTINm Input Pulse Interval Measurement Function . | Channel operation is stopped. |
| | TAUDnCDRm register operates as a capture register. | |
| Restart | Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000 _H . INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1. |
| During Operation | Detection of TAUDnTTINm edge The TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV can be written to 1. (TAUDnCSRm.TAUDnOVF bit is cleared to 0.) | TAUDnCNTm starts to count up from 0000 _H . When a TAUDnTTINm valid edge is detected: <ul style="list-style-type: none"> • TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and returns to 0000_H. • INTTAUDnIm is then generated. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values. |

23.12.7.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

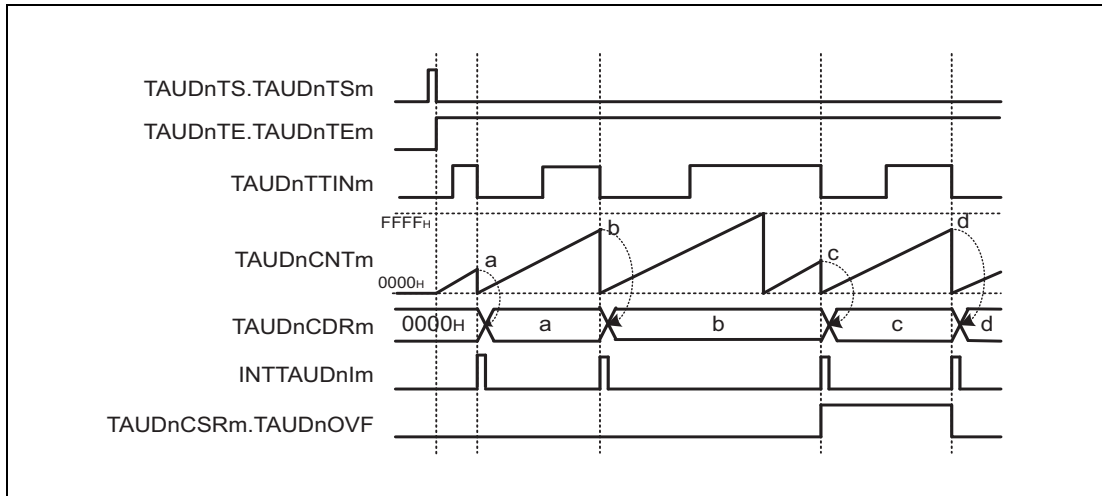


Figure 23.53 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains = 0.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01_B

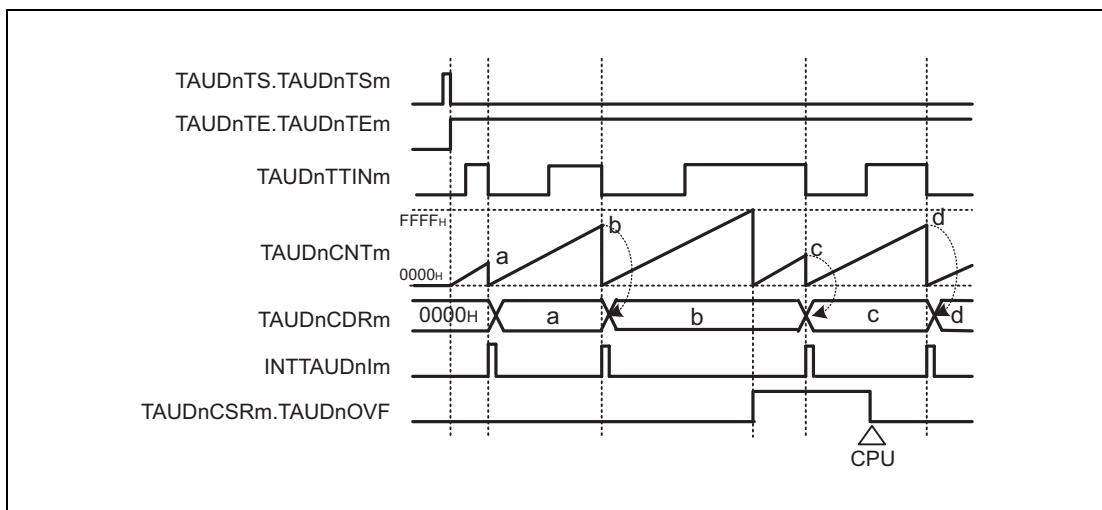


Figure 23.54 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

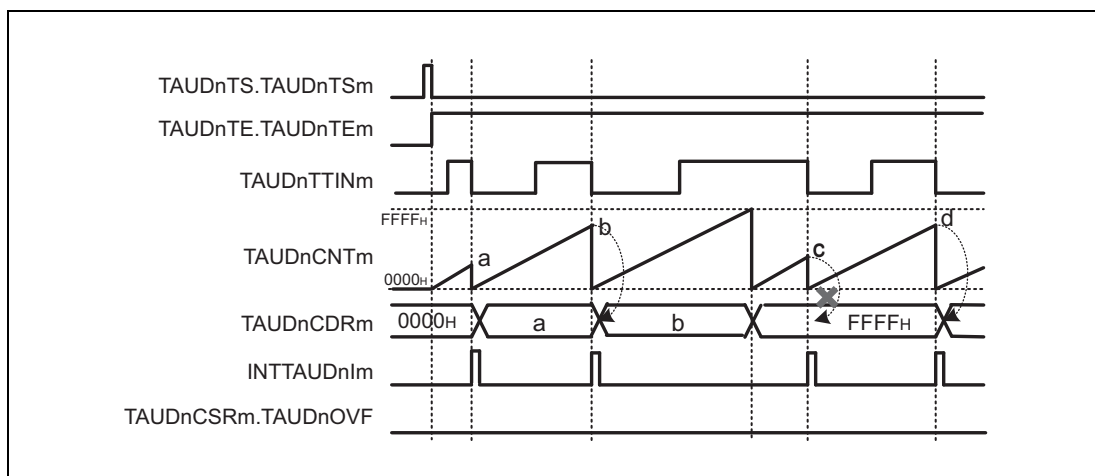


Figure 23.55 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to FFFF_H and TAUDnCSRm.TAUDnOVF remains = 0.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.

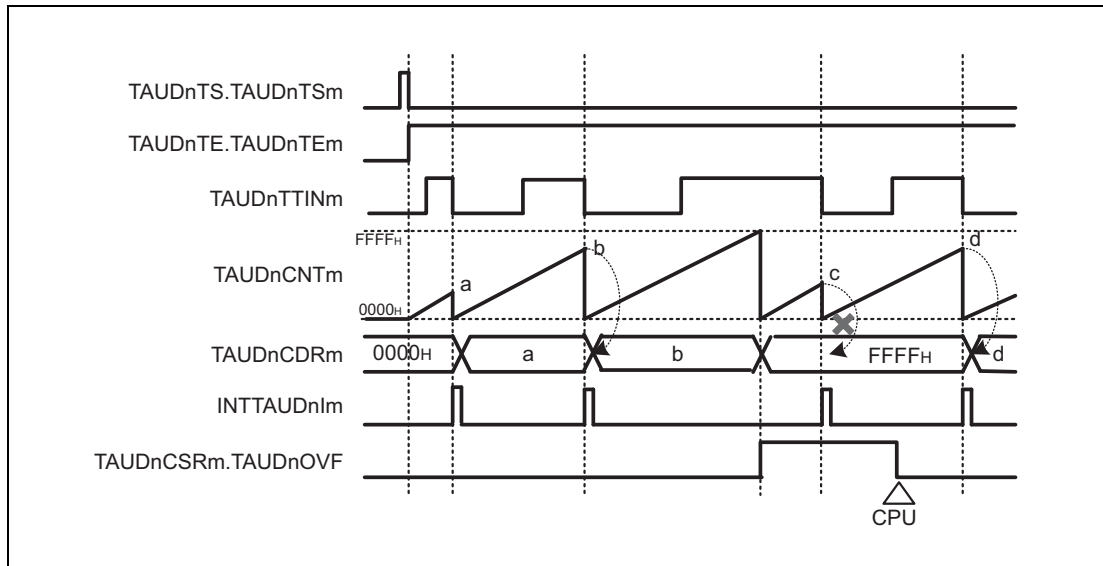
(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

Figure 23.56 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to FFFF_H and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

23.12.8 TAUDnTTINm Input Signal Width Measurement Function

23.12.8.1 Overview

Summary

This function measures the width of a TAUDnTTINm signal, by starting the count at one edge of TAUDnTTINm and capturing the count value at the other edge.

Prerequisites

- The operating mode should be set to capture and one-count mode. See **Table 23.78, Contents of the TAUDnCMORm register for TAUDnTTINm Input Signal Width Measurement Function**.
- TAUDnTTOUTm is not used with this function.
- TAUDnCMORm.TAUDnMD0 should be set to 0.

Functional description

The counter is started by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When a valid TAUDnTTINm start edge is detected, the counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDnTTINm stop edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter retains its value (CDRn + 1) and awaits the next valid TAUDnTTINm input start edge.

If the counter reaches FFFF_H before a valid TAUDnTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 23.77 Effects of Overflow

| TAUDnCMORm. TAUDnCOS[1:0] | When Overflow Occurs | | When a Valid TAUDnTTINm Input Stop Edge is Detected | |
|------------------------------|--------------------------|------------------------|---|------------------------|
| | TAUDnCDRm | TAUDnCSRm. TAUDnOVF | TAUDnCDRm, TAUDnCNTm | TAUDnCSRm. TAUDnOVF |
| 00 | Unchanged | 0 | TAUDnCNTm loaded into TAUDnCDRm | 1 |
| 01 | | 1 | | |
| 10 | Set to FFFF _H | 0 | TAUDnCNTm stops counting TAUDnCDRm unchanged | Unchanged |
| 11 | | 1 | | |

When TAUDnCMORm.TAUDnCOS[0] = 1, overflow bit TAUDnCSRm.TAUDnOVF can be cleared only by setting TAUDnCSCm.TAUDnCLOV to 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the width of the TAUDnTTINm signal. However, if an overflow occurs multiple times before a valid TAUDnTTINm input is detected, overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

This function cannot be forcibly restarted.

NOTE

When $TAUDnCMORm.TAUDnCOS[1] = 1$, the value of $TAUDnCNTm$ is not loaded to $TAUDnCDRm$ when the first valid $TAUDnTTINm$ input edge occurs after an overflow. However, an interrupt is generated.

23.12.8.2 Equations

$$TAUDnTTINm \text{ input signal width} = \text{count clock cycle} \times [(TAUDnCSRm.TAUDnOVF \times (FFFF_H + 1)) + TAUDnCDRm \text{ capture value} + 1]$$

23.12.8.3 Block Diagram and General Timing Diagram

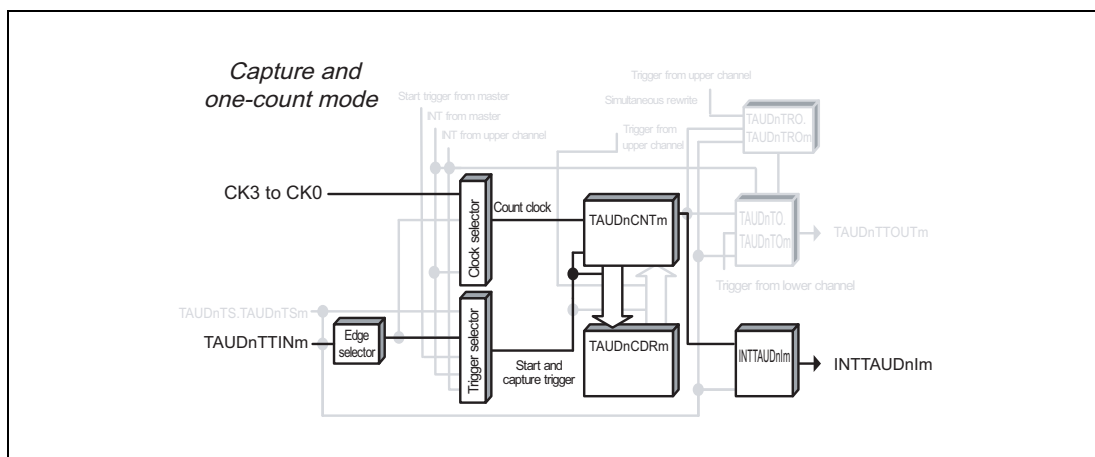


Figure 23.57 Block Diagram of TAUDnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement
(TAUDnCMURm.TAUDnTIS[1:0] = 11_B)
- When a valid TAUDnTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1. (TAUDnCMORm.TAUDnCOS[1:0] = 00_B)

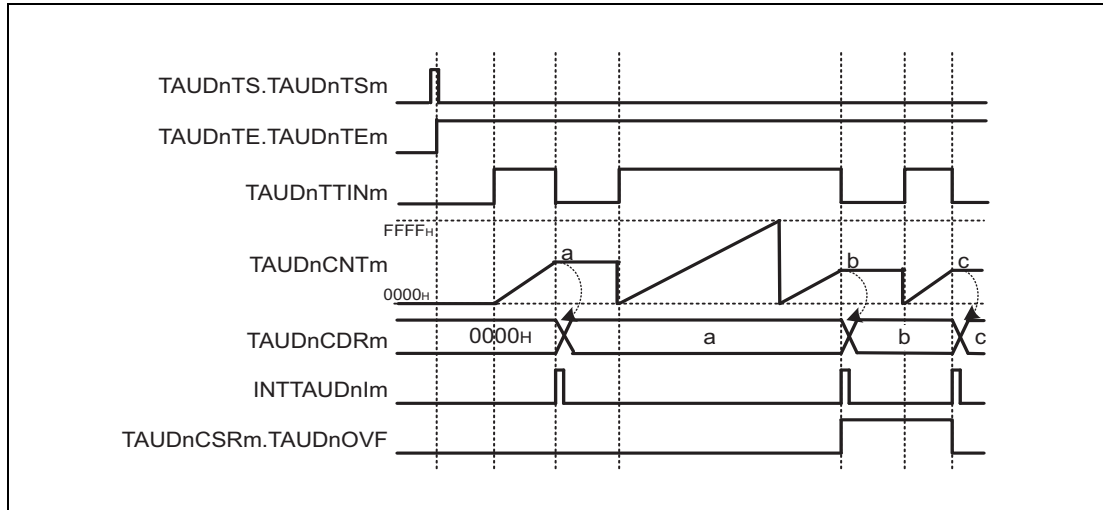


Figure 23.58 General Timing Diagram of TAUDnTTINm Input Signal Width Measurement Function

23.12.8.4 Register Settings

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.78 Contents of the TAUDnCMORm register for TAUDnTTINm Input Signal Width Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: prescaler output CK0 01: prescaler output CK1 10: prescaler output CK2 11: prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Independent operation. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger. |
| 7, 6 | TAUDnCOS[1:0] | See Table 23.77, Effects of Overflow. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0110: Capture and one-count mode |
| 0 | TAUDnMD0 | 0: Disables the start trigger during operation. |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.79 Contents of the TAUDnCMURm register for TAUDnTTINm Input Signal Width Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement) |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input signal width measurement function. Therefore, these registers should be set to 0.

Table 23.80 Simultaneous Rewrite Settings for TAUDnTTINm Input Signal Width Measurement Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.8.5 Operating Procedure for TAUDnTTINm Input Signal Width Measurement Function

Table 23.81 Operating Procedure for TAUDnTTINm Input Signal Width Measurement Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.78, Contents of the TAUDnCMORm register for TAUDnTTINm Input Signal Width Measurement Function , and Table 23.79, Contents of the TAUDnCMURm register for TAUDnTTINm Input Signal Width Measurement Function . TAUDnCDRm register operates as a capture register. | Channel operation is stopped. |
| Start Operation | Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. When a TAUDnTTINm start edge is detected, TAUDnCNTm starts to count up. |
| During Operation | TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV bit can be set to 1. | TAUDnCNTm starts to count up from 0000 _H . When TAUDnTTINm valid edge is detected: <ul style="list-style-type: none"> TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and retains its value. INTTAUDnIm is then generated. Counting stops at the “value that transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values. |

Restart

23.12.8.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

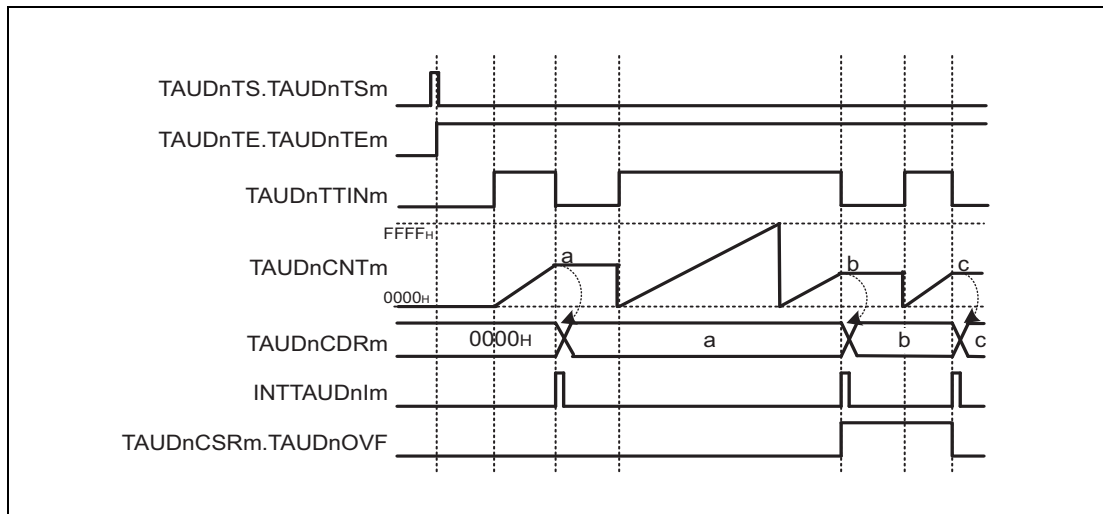


Figure 23.59 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains = 0.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

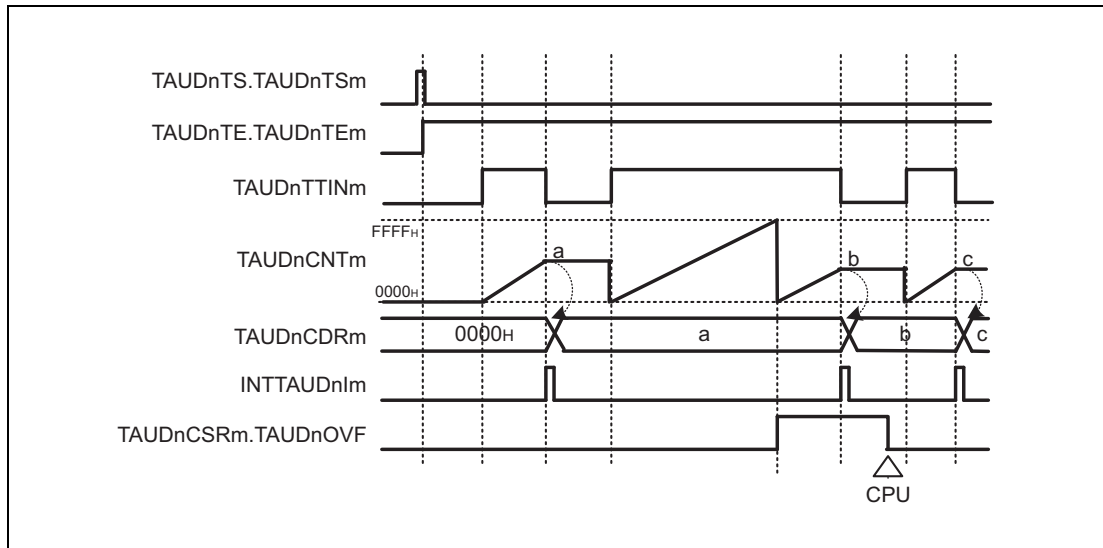
(2) TAUDnCMORm.TAUDnCOS[1:0] = 01_B

Figure 23.60 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command. (by setting TAUDnCSCm.TAUDnCLOV bit to 1.)

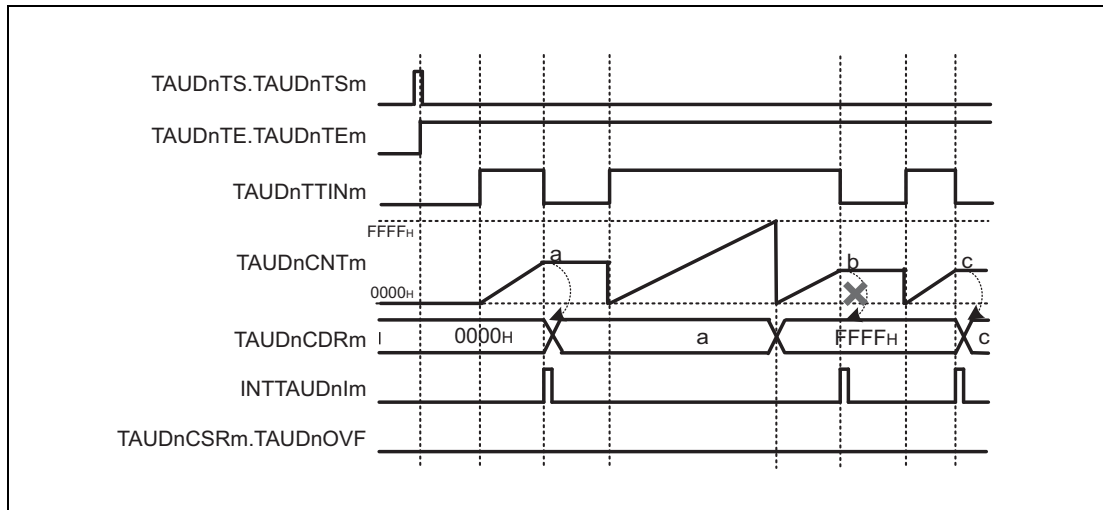
(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

Figure 23.61 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to FFFF_H and TAUDnCSRm.TAUDnOVF remains = 0.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm stops, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.

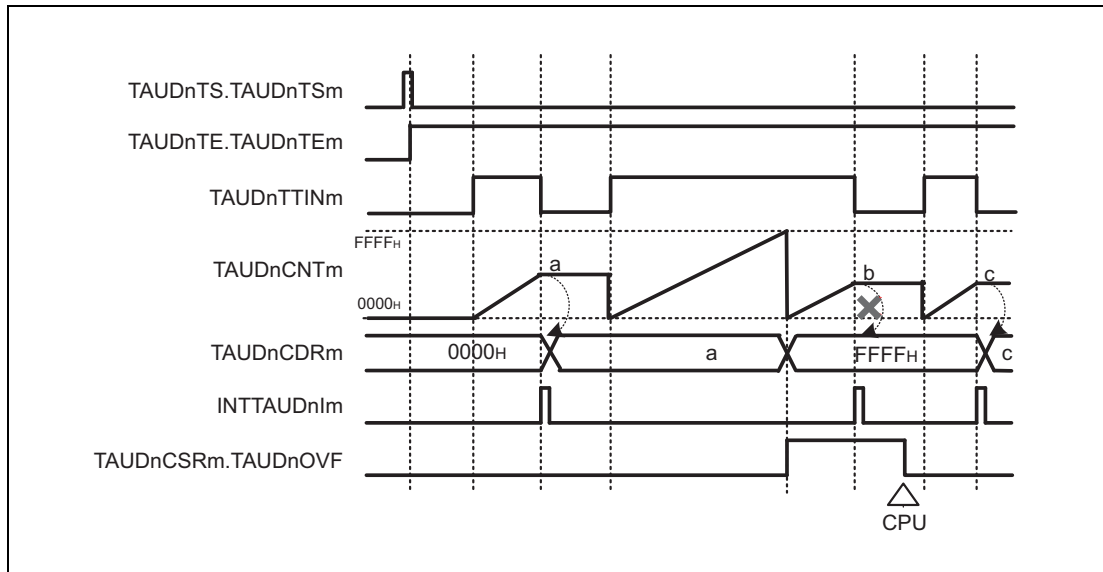
(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

Figure 23.62 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to FFFF_H and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm stops, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

23.12.9 TAUDnTTINm Input Position Detection Function

23.12.9.1 Overview

Summary

This function measures the input signal duration by capturing the count value at the valid edge of TAUDnTTINm.

Prerequisites

- The operating mode should be set to count capture mode. (See **Table 23.82, Contents of the TAUDnCMORm register for TAUDnTTINm Input Position Detection Function.**)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter starts counting from 0000_H. When a valid TAUDnTTINm input edge is detected, the current value of TAUDnCNTm is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The count operation continues.

When the counter reaches FFFF_H, the counter restarts to count from 0000_H.

NOTE

The TAUDTTINm input signal is sampled at the frequency of the operation clock specified by the TAUDnCMORm.TAUDnCKS[1:0] bits. As a result, the output cycle of the TAUDTTOUTm has an error of ± 1 operation clock cycle.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt does not occur at the beginning of operation or after restart. For details, see **Section 23.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

23.12.9.2 Equations

Functional duration at a TAUDnTTINm input pulse =
 count clock cycle \times (TAUDnCDRm capture value + 1)

23.12.9.3 Block Diagram and General Timing Diagram

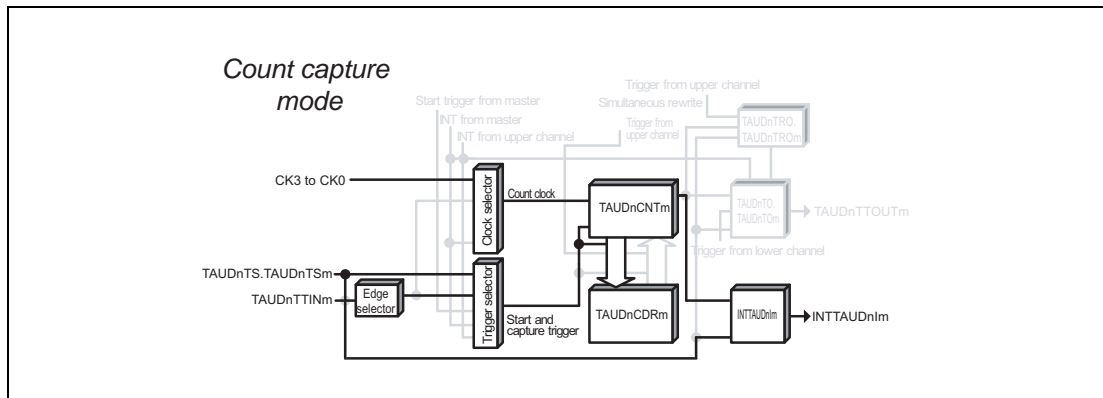


Figure 23.63 Block Diagram of TAUDnTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUDnIm not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

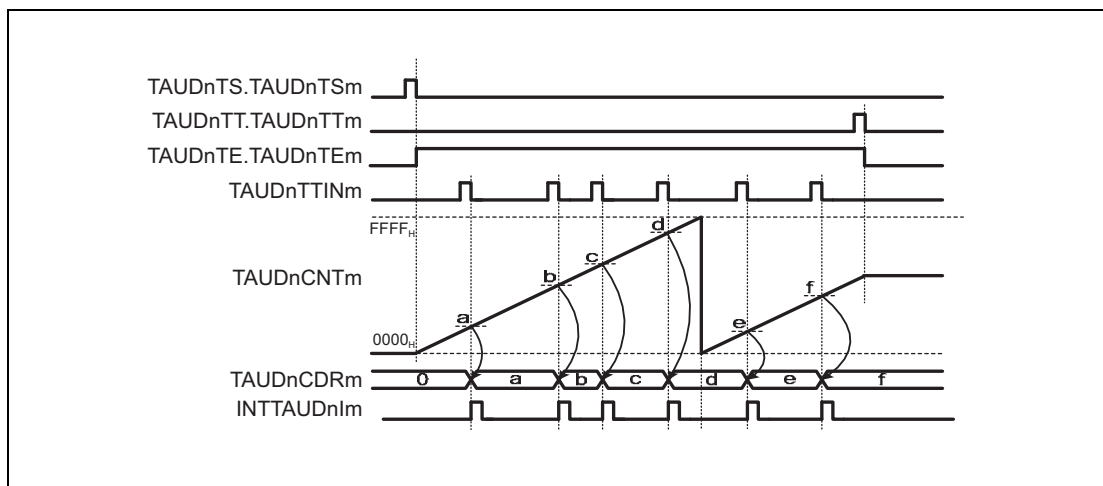


Figure 23.64 General Timing Diagram of TAUDnTTINm Input Position Detection Function

23.12.9.4 Register Settings

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.82 Contents of the TAUDnCMORm register for TAUDnTTINm Input Position Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid TAUDnTTINm input edge signal is used as an external capture trigger. |
| 7, 6 | TAUDnCOS[1:0] | 01: Set this value. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1011: Count capture mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.83 Contents of the TAUDnCMURm register for TAUDnTTINm Input Position Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges |

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with TAUDnTTINm input position detection function. Therefore, these registers should be set to 0.

Table 23.84 Simultaneous Rewrite Settings for TAUDnTTINm Input Position Detection Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.9.5 Operating Procedure for TAUDnTTINm Input Position Detection Function

Table 23.85 Operating Procedure for TAUDnTTINm Input Position Detection Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.82, Contents of the TAUDnCMORm register for TAUDnTTINm Input Position Detection Function , and Table 23.83, Contents of the TAUDnCMURm register for TAUDnTTINm Input Position Detection Function . TAUDnCDRm register operates as a capture register. | Channel operation is stopped. |
| Restart | Set TAUDnTS.TAUDnTSM to 1. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is set to 1 and the counter starts. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm occurs. |
| During Operation | The TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time. | TAUDnCNTm starts to count up from 0000 _H . When a valid TAUDnTTINm edge is detected: <ul style="list-style-type: none"> • TAUDnCNTm transfers (captures) its own value to TAUDnCDRm. • INTTAUDnIm occurs. • The counter is not cleared to 0000_H and TAUDnCNTm continues counting. Afterwards, this procedure is repeated. When TAUDnCNTm reaches FFFF _H , the counter restarts from 0000 _H . |
| Stop Operation | Set TAUDnTT.TAUDnTTM to 1. TAUDnTT.TAUDnTTM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm retains its current value. |

23.12.9.6 Specific Timing Diagrams

(1) Operation stop and restart

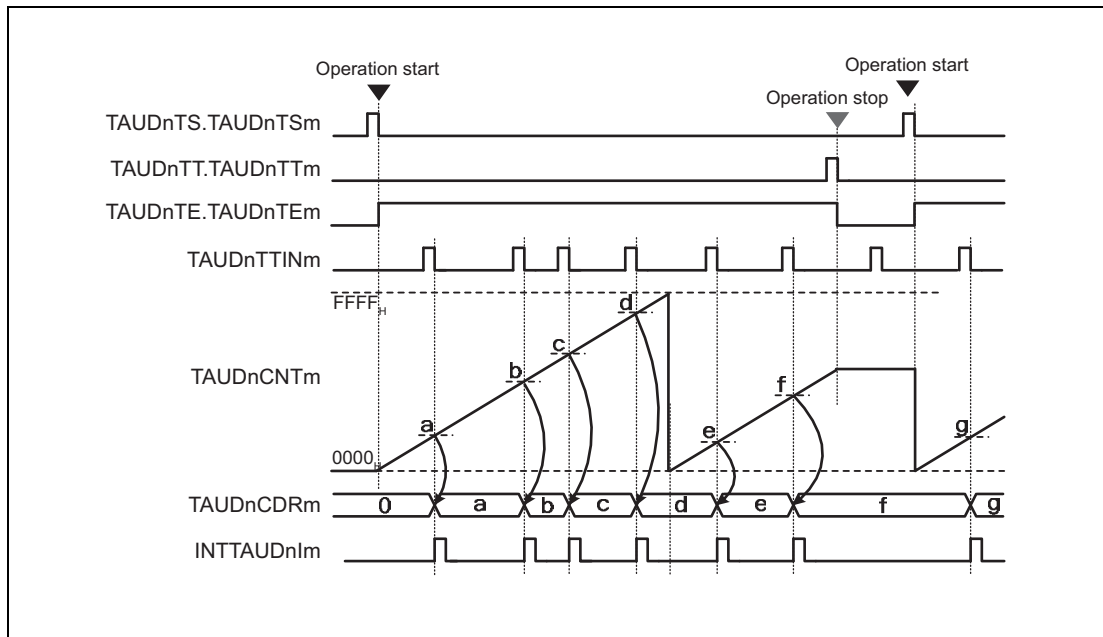


Figure 23.65 Operation Stop and Restart
 (TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- The counter can stop operating by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter stops operating, valid TAUDnTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSM to 1. TAUDnCNTm restarts to count from 0000_H.

23.12.10 TAUDnTTINm Input Period Count Detection Function

23.12.10.1 Overview

Summary

This function measures the cumulative width of a TAUDnTTINm input signal.

Prerequisites

- The operating mode should be set to capture and gate count mode. (See **Table 23.86, Contents of the TAUDnCMORm register for TAUDnTTINm Input Period Count Detection Function.**)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The counter awaits a valid TAUDnTTINm input edge.

When a valid TAUDnTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUDnTTINm input stop edge is detected, the current TAUDnCNTm value is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The counter stops and retains its value (TAUDnCDRm + 1) until the next valid TAUDnTTINm input start edge is detected.

When the next valid TAUDnTTINm input start edge is detected, the counter restarts to count from the value retained when stopped.

If the counter reaches FFFF_H, the counter restart to count from 0000_H.

NOTES

1. TAUDnTTINm input signal is sampled at the frequency of an operating clock set by the TAUDnCMORm.TAUDnCKS[1:0] bits.
2. As this function is to measure the TAUDnTTINm input signal width, setting TAUDnTS.TAUDnTSM to 1 is disabled while TAUDnTE.TAUDnTEM = 1.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits:

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDnTTINm input low period is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDnTTINm input high period is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

23.12.10.2 Equations

Cumulative TAUDnTTINm input width = count clock cycle × (TAUDnCDRm capture value + 1)

23.12.10.3 Block Diagram and General Timing Diagram

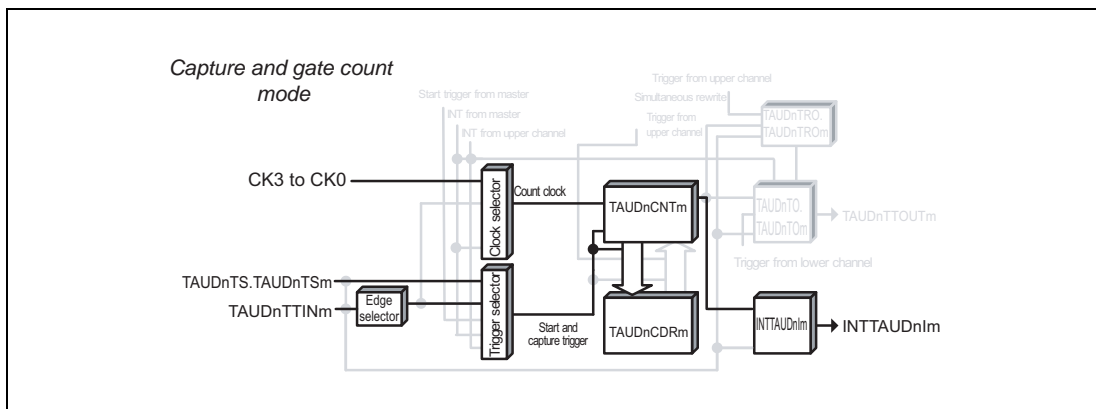


Figure 23.66 Block Diagram of TAUDnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement
(TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

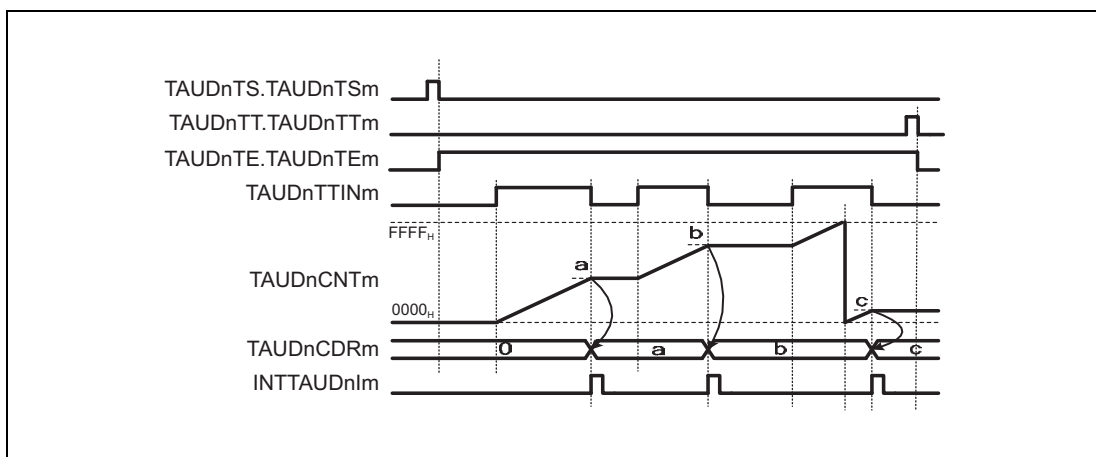


Figure 23.67 General Timing Diagram of TAUDnTTINm Input Period Count Detection Function

23.12.10.4 Register Settings

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.86 Contents of the TAUDnCMORm register for TAUDnTTINm Input Period Count Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger. |
| 7, 6 | TAUDnCOS[1:0] | 01: Set to this value. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1101: Capture and gate count mode |
| 0 | TAUDnMD0 | 0: Disables the start trigger during operation. |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.87 Contents of the TAUDnCMURm register for TAUDnTTINm Input Period Count Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement) |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with TAUDnTTINm input period count detection. Therefore, these registers should be set to 0.

Table 23.88 Simultaneous Rewrite Settings for TAUDnTTINm Input Period Count Detection Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.10.5 Operating Procedure for TAUDnTTINm Input Period Count Detection Function

Table 23.89 Operating Procedure for TAUDnTTINm Input Period Count Detection Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.86, Contents of the TAUDnCMORm register for TAUDnTTINm Input Period Count Detection Function , and Table 23.87, Contents of the TAUDnCMURm register for TAUDnTTINm Input Period Count Detection Function . | Channel operation is stopped. |
| | TAUDnCDRm register operates as a capture register. | |
| Restart | Set TAUDnTS.TAUDnTSM to 1. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. |
| | Detection of TAUDnTTINm edge The TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time. | When a TAUDnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUDnCNTm starts counting up from the stop value. When TAUDnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUDnCDRm and INTTAUDnIm is generated. Counting stops at the "value transferred to TAUDnCDRm + 1" and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. When TAUDnCNTm reaches FFFF _H , the counter restarts to count from 0000 _H . Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm retains its current value. |

23.12.10.6 Specific Timing Diagrams

(1) Operation stop and restart

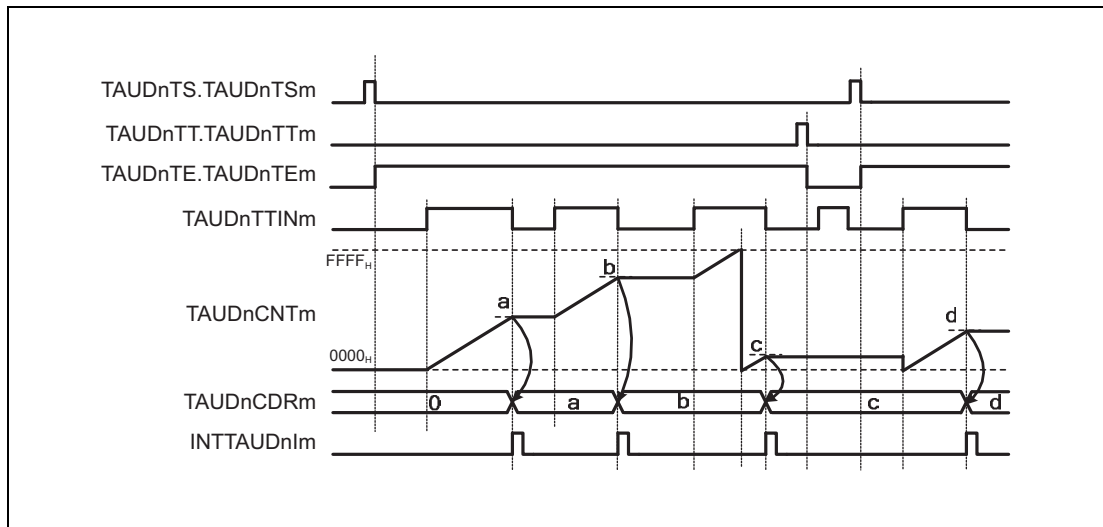


Figure 23.68 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter is stopped, valid TAUDnTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSM to 1. TAUDnCNTm restarts to count from 0000_H.

23.12.11 TAUDnTTINm Input Pulse Interval Judgment Function

23.12.11.1 Overview

Summary

This function outputs the result of a comparison between the count value (TAUDnCNTm) and the value in the channel data register (TAUDnCDRm) when a TAUDnTTINm input pulse occurs. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true.

Prerequisites

- The operating mode should be set to judge mode. See **Table 23.90, Contents of the TAUDnCMORm register for TAUDnTTINm Input Pulse Interval Judgment Function.**
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is started by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDnTTINm valid edge is detected or TAUDnTS.TAUDnTSM is set to 1, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. TAUDnCNTm reloads the value of TAUDnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUDnTTINm valid edge is detected, TAUDnCNTm overflows and is set to FFFF_H. It then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:

- If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when $\text{TAUDnCNTm} \leq \text{TAUDnCDRm}$.
- If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when $\text{TAUDnCNTm} > \text{TAUDnCDRm}$.

23.12.11.2 Block Diagram and General Timing Diagram

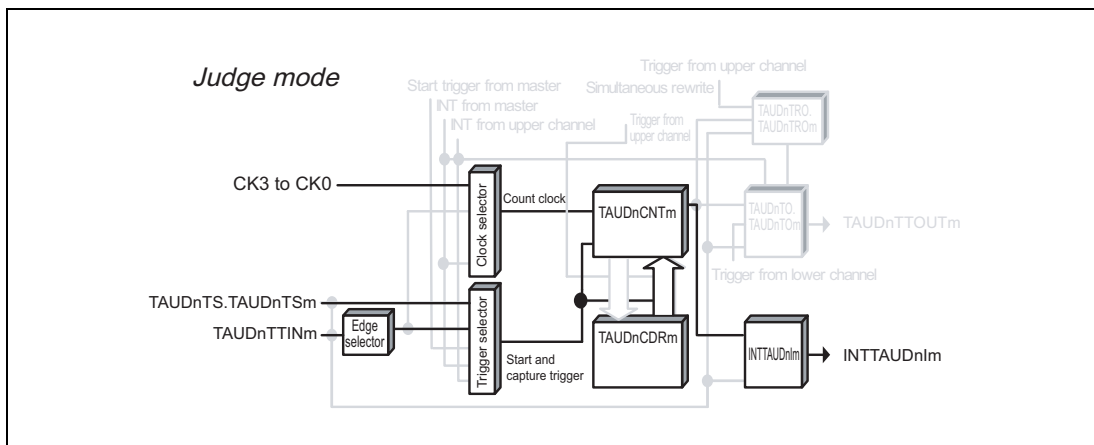


Figure 23.69 Block Diagram of TAUDnTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

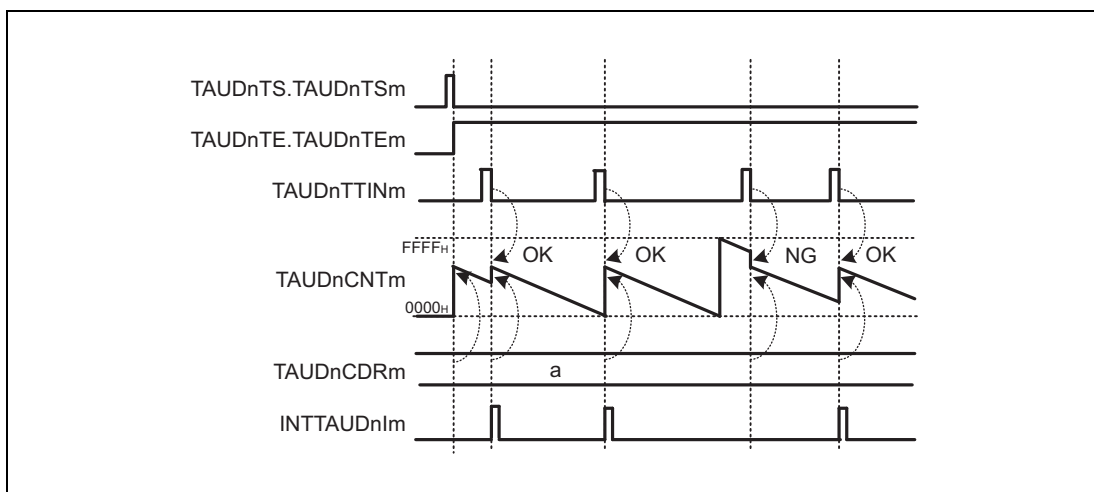


Figure 23.70 General Timing Diagram of TAUDnTTINm Input Pulse Interval Judgment Function

23.12.11.3 Register Settings

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.90 Contents of the TAUDnCMORm register for TAUDnTTINm Input Pulse Interval Judgment Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Independent operation. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid edge of the TAUDnTTINm input signal is used as an external start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0001: Judge mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.91 Contents of the TAUDnCMURm register for TAUDnTTINm Input Pulse Interval Judgment Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with TAUDnTTINm input pulse interval judgment. Therefore, these registers should be set to 0.

Table 23.92 Simultaneous Rewrite Settings for TAUDnTTINm Input Pulse Interval Judgment Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.11.4 Operating Procedure for TAUDnTTINm Input Pulse Interval Judgment Function

Table 23.93 Operating Procedure for TAUDnTTINm Input Pulse Interval Judgment Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.90, Contents of the TAUDnCMORm register for TAUDnTTINm Input Pulse Interval Judgment Function , and Table 23.91, Contents of the TAUDnCMURm register for TAUDnTTINm Input Pulse Interval Judgment Function . | Channel operation is stopped. |
| | Set the value of TAUDnCDRm register. | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm. |
| During Operation | The following register can be changed at any time: <ul style="list-style-type: none"> TAUDnCNTm register | When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDnTTINm input edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDnTTINm input edge is detected, INTTAUDnIm is generated. If a TAUDnTTINm input edge is detected, then TAUDnCNTm starts to count down from the value of TAUDnCDRm. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value. |

23.12.12 TAUDnTTINm Input Signal Width Judgment Function

23.12.12.1 Overview

Summary

This function compares the count value (TAUDnCNTm) for the high or low level width of a TAUDnTTINm input signal and the TAUDnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUDnIm.

Prerequisites

- The operating mode should be set to judge and one-count mode. (See **Table 23.94, Contents of the TAUDnCMORm register for TAUDnTTINm Input Signal Width Judgment Function**)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is started by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. When a valid TAUDnTTINm input start edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDnTTINm valid stop edge is detected, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. The counter TAUDnCNTm retains its value until the next valid TAUDnTTINm start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUDnTTINm stop edge is detected, TAUDnCNTm overflows and is set to FFFF_H. The counter then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

- The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:
 - If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when $TAUDnCNTm \leq TAUDnCDRm$.
 - If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when $TAUDnCNTm > TAUDnCDRm$.
- The TAUDnCMURm.TAUDnTIS[1:0] bits specify a type of width measurement:
 - For high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B), TAUDnTTINm rising edge is used as a start edge and TAUDnTTINm falling edge as a stop edge.
 - For low width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 10_B), TAUDnTTINm falling edge is used as a start edge and TAUDnTTINm rising edge as a stop edge.
- This function cannot make a forced restart.

23.12.12.2 Block Diagram and General Timing Diagram

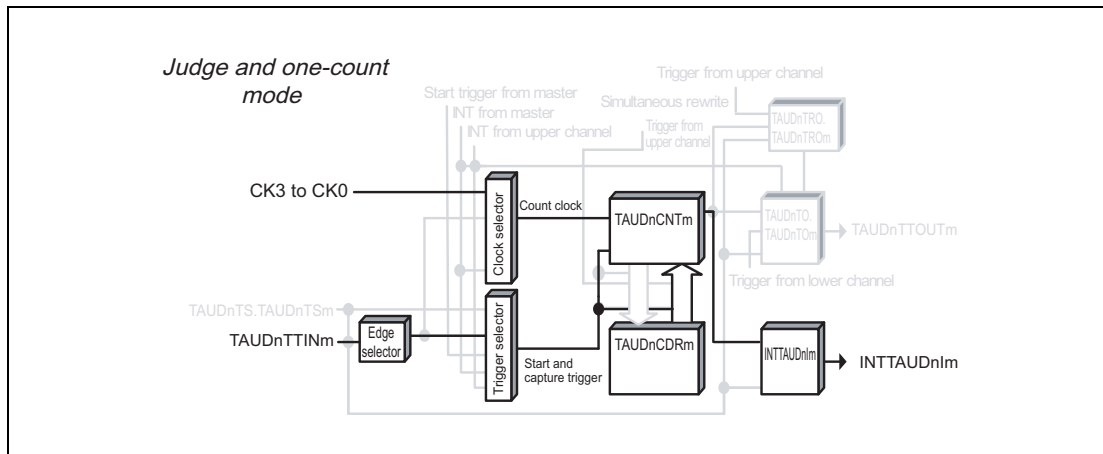


Figure 23.71 Block Diagram of TAUDnTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated when $TAUDnCNTm \leq TAUDnCDRm$ ($TAUDnCMORm.TAUDnMD0 = 0$)
- TAUDnTTINm valid start edge = rising edge, TAUDnTTINm valid stop edge = falling edge ($TAUDnCMURm.TAUDnTIS[1:0] = 11_B$)

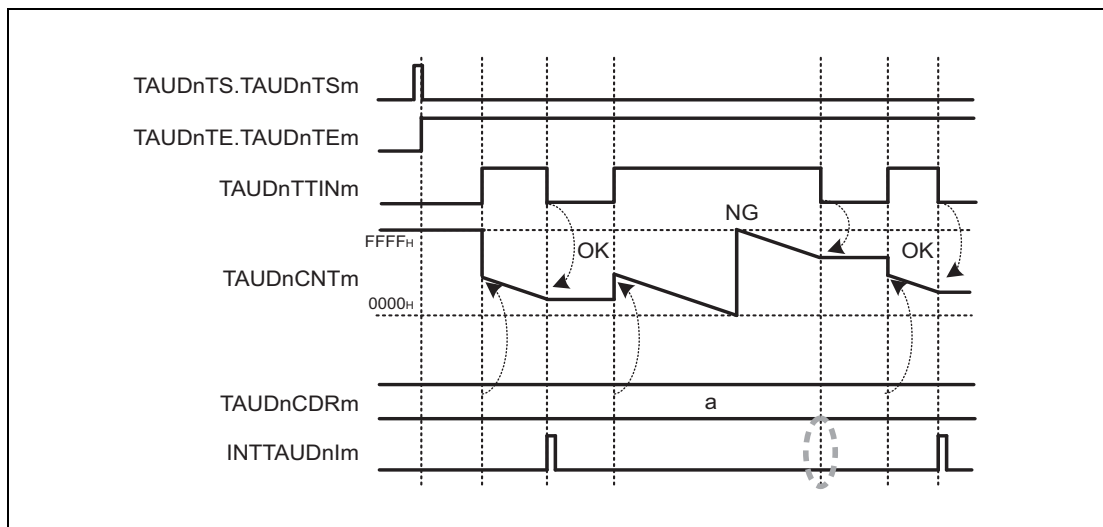


Figure 23.72 General Timing Diagram of TAUDnTTINm Input Signal Width Judgment Function

23.12.12.3 Register Settings

(1) TAUDnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.94 Contents of the TAUDnCMORm register for TAUDnTTINm Input Signal Width Judgment Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger. |
| 7, 6 | TAUDnCOS[1:0] | 0: Independent operation. Set to 0. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0111: Judge and one-count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm |

(2) TAUDnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.95 Contents of the TAUDnCMURm register for TAUDnTTINm Input Signal Width Judgment Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement) |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with TAUDnTTINm input signal width judgment. Therefore, these registers should be set to 0.

Table 23.96 Simultaneous Rewrite Settings for TAUDnTTINm Input Signal Width Judgment Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.12.4 Operating Procedure for TAUDnTTINm Input Signal Width Judgment Function

Table 23.97 Operating Procedure for TAUDnTTINm Input Signal Width Judgment Function

| | Operation | TAUDn Status |
|-------------------------|---|--|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers as described in Table 23.94, Contents of the TAUDnCMORm register for TAUDnTTINm Input Signal Width Judgment Function , and Table 23.95, Contents of the TAUDnCMURm register for TAUDnTTINm Input Signal Width Judgment Function . | Channel operation is stopped. |
| | Set the value of TAUDnCDRm register. | |
| Start Operation | Set TAUDnTS.TAUDnTSM to 1. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. |
| | The following register can be changed at any time: • TAUDnCDRm register | Upon detection of a TAUDnTTINm start edge, TAUDnCNTm starts count down from the value of TAUDnCDRm. |
| | | When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a INTTAUDnIm input stop edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDnTTINm input stop edge is detected, INTTAUDnIm is generated. |
| During Operation | | Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTM to 1. TAUDnTT.TAUDnTTM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value. |



23.12.13 One-Phase PWM Output Function

23.12.13.1 Overview

Summary

This function adds dead time to a TAUDnTTINm input signal. The resulting PWM signal is output via TAUDnTTOUTm of the channel and TAUDnTTOUTm of upper channels.

Prerequisites

- Each of two (or more) channels is enabled for dead time control (TAUDnTDE.TAUDnTDEm = 1).
- The operating mode for the lower channel should be set to one-count mode. (See **Table 23.99, Contents of the TAUDnCMORm register for the Lower Channel of the One-Phase PWM Output Function.**)
- Any operating mode can be set to upper channels.
- Channel output mode for upper and lower channels should be set to synchronous channel output mode 2 with one-phase PWN output. (See **Section 23.7, Channel Output Modes.**)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt occurs. The counter is reset to FFFF_H and waits for the next valid TAUDnTTINm input start edge.

Table 23.98 TAUDnTTOUTm to which Dead Time is Added and State of TAUDnTTINm

| TAUDnCMURm. TAUDnTIS[1:0] | TAUDnTOL. TAUDnTOLm | TAUDnTTOUTm to which Dead Time is Added | TAUDnTDL. TAUDnTDLm | TAUDnTTINm State When Added |
|------------------------------|------------------------|--|------------------------|--------------------------------|
| 10 | 0 | TAUDnTTOUTm low | 0 | High |
| | | | 1 | Low |
| | 1 | TAUDnTTOUTm high | 0 | High |
| | | | 1 | Low |
| 11 | 0 | TAUDnTTOUTm low | 0 | Low |
| | | | 1 | High |
| | 1 | TAUDnTTOUTm high | 0 | Low |
| | | | 1 | High |

Conditions

- TAUDnCMURm.TAUDnTIS[1:0] bits specify the type of width measurement:
 - TAUDnCMURm.TAUDnTIS[1:0] = 10_B: Uses both edges as valid edges for detection (Low width measurement).
 - TAUDnCMURm.TAUDnTIS[1:0] = 11_B: Uses both edges as valid edges for detection (High width measurement).
- The TAUDnTDL.TAUDnTDLm bit specifies the operation of TAUDnTTOUTm for each channel when an interrupt or valid TAUDnTTINm edge is detected on the lower channel:
 - If TAUDnTDL.TAUDnTDLm = 0, dead time to the positive-phase width (normal) is added.
 - If TAUDnTDL.TAUDnTDLm = 1, dead time to the negative-phase width (inverted) is added.
- This function cannot make a forced restart.

23.12.13.2 Block Diagram and General Timing Diagram

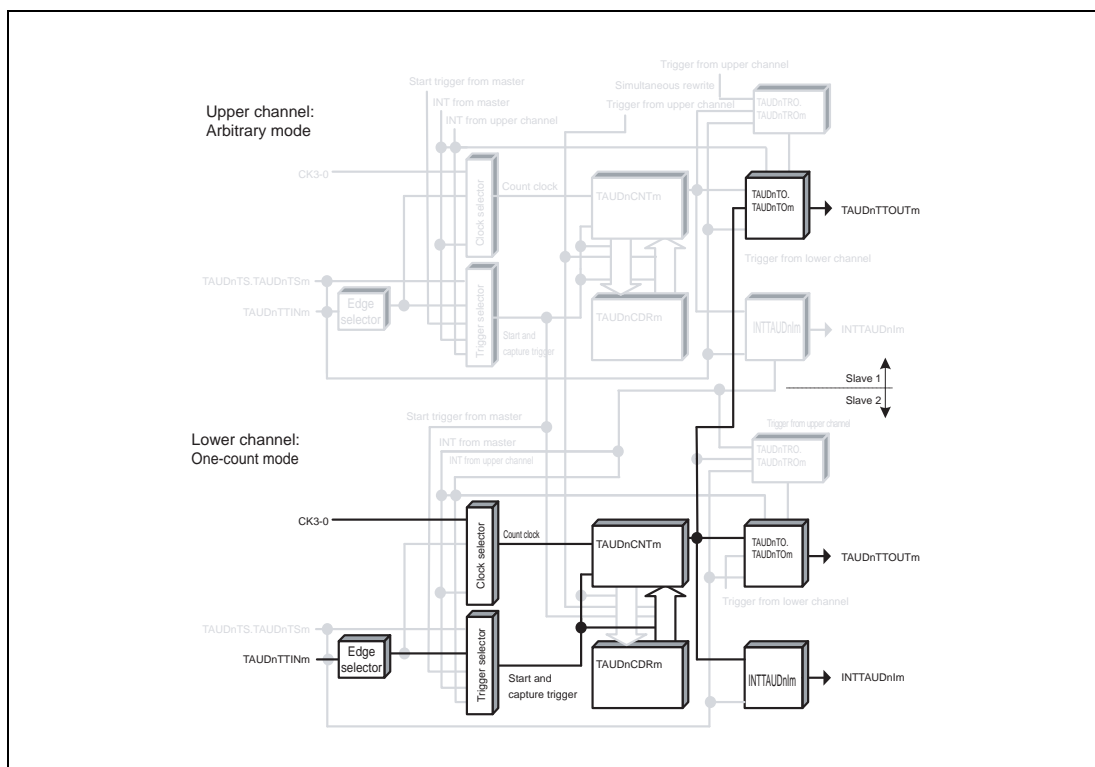


Figure 23.73 Block Diagram of One-Phase PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement
(TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

This setting considers a duty as a high active.

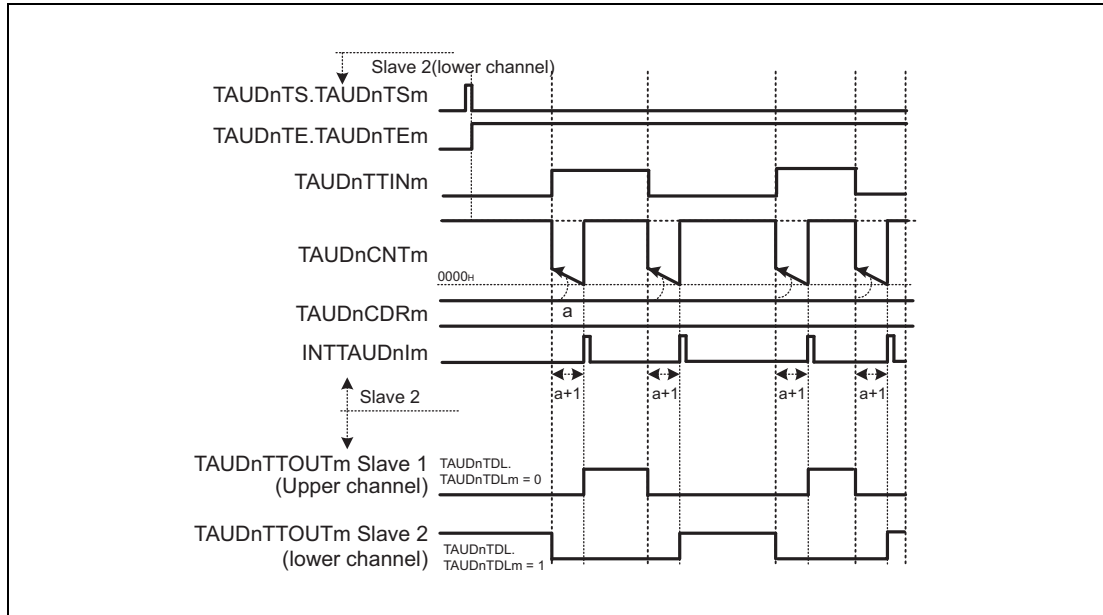


Figure 23.74 General Timing Diagram of One-Phase PWM Output Function

23.12.13.3 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.99 Contents of the TAUDnCMORm register for the Lower Channel of the One-Phase PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid edge of the TAUDnTTINm input signal is used as an external start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Enables start trigger detection while counting. |

(2) TAUDnCMURm for lower channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.100 Contents of the TAUDnCMURm register for the Lower Channel of the One-Phase PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement) |

(3) Channel output mode for lower channels**Table 23.101 Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output**

| Bit Name | Setting |
|--------------------|--|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 1: Enables dead time operation. |
| TAUDnTDM.TAUDnTDMm | 1: Adds dead time upon detection of a TAUDnTTINm input edge on a lower odd-numbered channel. |
| TAUDnTDL.TAUDnTDLm | 0: Adds dead time to the positive-phase width (normal) 1: Adds dead time to the negative-phase width (inverted) |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROM | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

CAUTION

Set TAUDnTDL.TAUDnTDLm to upper channels exclusively.

(4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 23.102 Simultaneous Rewrite Settings for One-Phase PWM Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.13.4 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

TAUDnCMORm register for upper channels can be set arbitrarily.

(2) TAUDnCMURm for upper channels

TAUDnCMURm register for upper channels can be set arbitrarily.

(3) Channel output mode for upper channels

Table 23.103 Control Bit Settings for Upper Channels in Synchronous Channel Output Mode 2 with One-Phase PWM Output

| Bit Name | Setting |
|--------------------|--|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 1: Enables dead time operation. |
| TAUDnTDM.TAUDnTDMm | 1: Adds dead time upon detection of a TAUDnTTINm input edge on a lower odd-numbered channel. |
| TAUDnTDL.TAUDnTDLm | 0: Adds dead time to the normal-phase width 1: Adds dead time to the inverted-phase width |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

CAUTION

Set TAUDnTDL.TAUDnTDLm to lower channels exclusively.

(4) Simultaneous rewrite for upper channels

Simultaneous rewrite register for upper channels can be set arbitrarily.

23.12.13.5 Operating Procedure for One-phase PWM Output Function

Table 23.104 Operating Procedure for One-phase PWM Output Function

| | Operation | TAUDn Status |
|-------------------------|--|--|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in Table 23.99, Contents of the TAUDnCMORm register for the Lower Channel of the One-Phase PWM Output Function , and Table 23.100, Contents of the TAUDnCMURm register for the Lower Channel of the One-Phase PWM Output Function . | Channel operation is stopped. |
| | Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in Section 23.12.13.4, Register Settings for Upper Channels . | |
| | Set the value of TAUDnCDRm register. | |
| | Set channel output mode by setting the control bits as described in Table 23.101, Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output . | |
| Restart | Start Operation | |
| | <p>Set TAUDnTOE.TAUDnTOEm (slave channels 1 and 2) to 1 (at restart time only). Set TAUDnTS.TAUDnTSm = 1 for slave channel 2. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDnTTINm start edge</p> | <p>TAUDnTE.TAUDnTEm is set to 1 (slave channel 2) and TAUDnCNTm waits for detection of TAUDnTTINm start edge. TAUDnCNTm loads TAUDnCDRm value.</p> |
| During Operation | | |
| | <p>The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at any time.</p> | <p>TAUDnCNTm of slave channel 2 counts down. When the counter reaches 0000_H:</p> <ul style="list-style-type: none"> • INTTAUDnIm is generated. • TAUDnCNTm stops counting. <p>TAUDnTTOUTm is changed by a TAUDnTTINm edge detection signal and slave channel 2 INTTAUDnIm signal to output one-phase PWM waveform with dead time. Afterwards, this operation is repeated.</p> |
| Stop Operation | <p>Set TAUDnTT.TAUDnTTm = 1 for slave channel 2. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p> | <p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDnTTOUTm retain their current values.</p> |

23.12.14 Real Time Output Function Type 1

23.12.14.1 Overview

Summary

This function outputs a value of the TAUDnTRO.TAUDnTROm bit from TAUDnTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDnTTOUTm control of other channels.
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 23.105, Contents of the TAUDnCMORM register for the Upper Channel of Real-Time Output Function Type 1.**)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 23.7, Channel Output Modes.**)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREM = 1).

Functional description

The counter of the upper channel is started by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The current value of the data register of the upper channel (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000_H, INTTAUDnIm is generated and TAUDnTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREM = 1). TAUDnCNTm then reloads the TAUDnCDRm value to continue operation subsequently.

The TAUDnTTOUTm signal changes only when an interrupt is generated, and when its value is different to current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREM = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.

- If real-time output of a lower channel is enabled ($TAUDnTRE.TAUDnTREM = 1$) and $TAUDnTRC.TAUDnTRCm = 0$, the value of that channel's $TAUDnTRO.TAUDnTROm$ bit is output when $INTTAUDnIm$ is generated in the upper channel.
- If the $TAUDnCMORm.TAUDnMD0$ bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 23.9, $TAUDnTTOUTm$ Output and $INTTAUDnIm$ Generation when Counter Starts or Restarts.**

23.12.14.2 Equations

$$INTTAUDnIm \text{ generation cycle} = \text{count clock cycle} \times (TAUDnCDRm \text{ value} + 1)$$

23.12.14.3 Block Diagram and General Timing Diagram

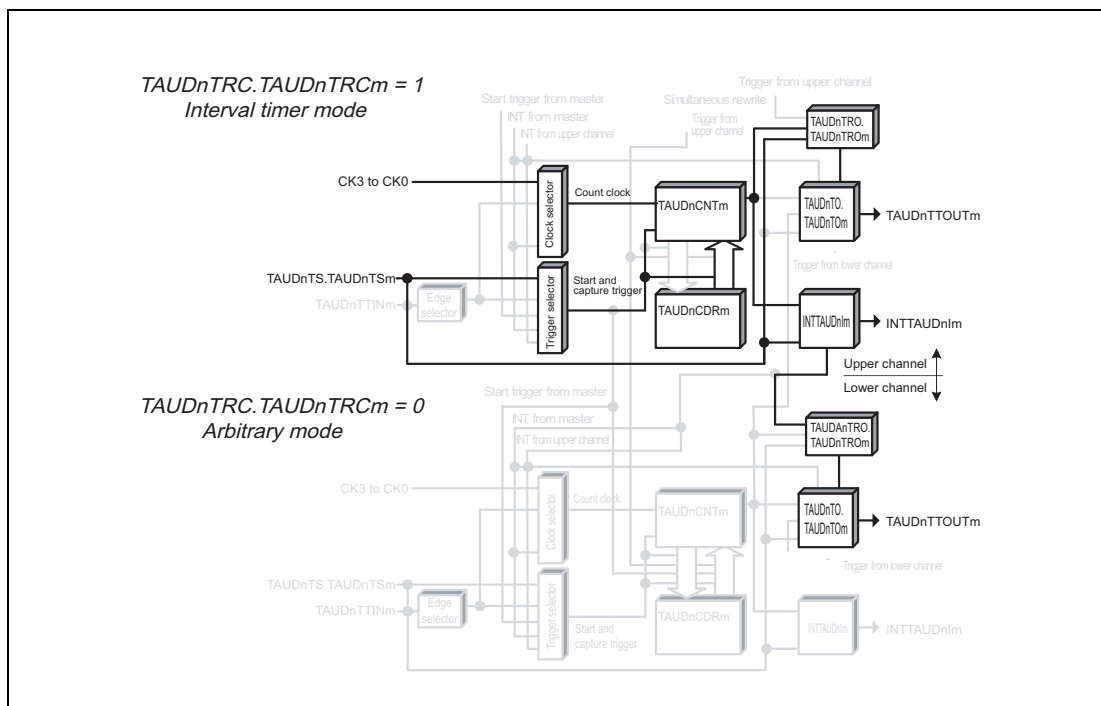


Figure 23.75 Block Diagram of Real-Time Output Function Type 1

The following settings apply to the general timing diagram.

- $INTTAUDnIm$ generated at the beginning of operation. ($TAUDnCMORm.TAUDnMD0 = 1$)

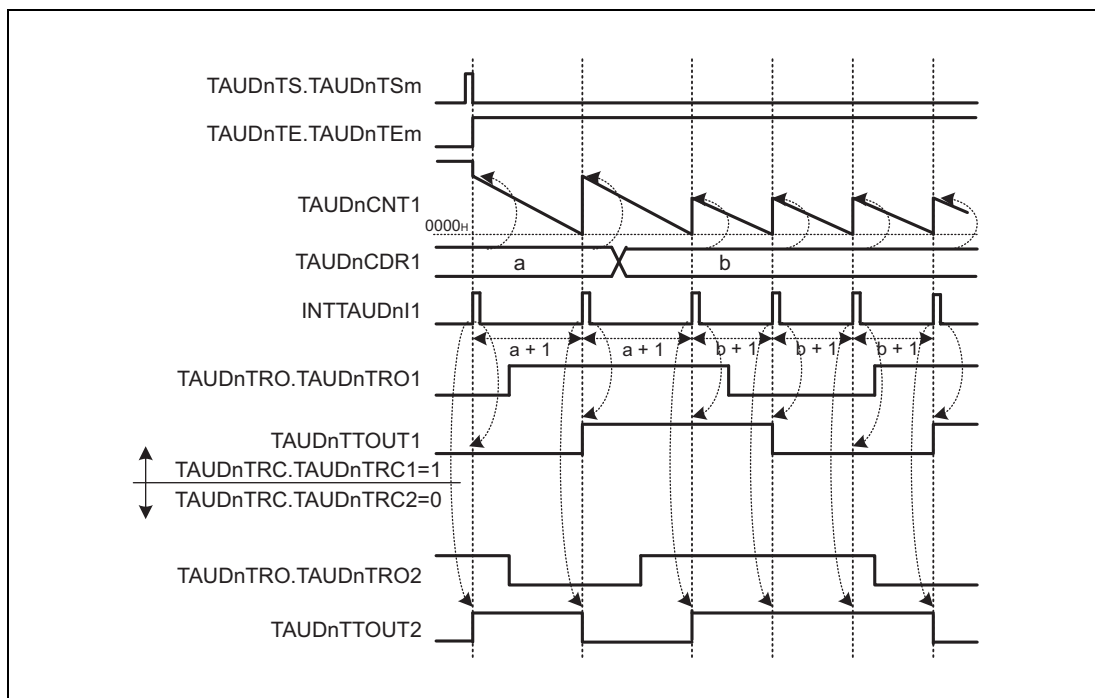


Figure 23.76 General Timing Diagram of Real-Time Output Function Type 1

23.12.14.4 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.105 Contents of the TAUDnCMORm register for the Upper Channel of Real-Time Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|-------------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[2:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS [1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm for upper channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.106 Contents of the TAUDnCMURm register for Upper Channel of Real-Time Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for upper channels**Table 23.107 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: Setting is disabled (value after reset) in toggle mode. |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROm | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 1: Channel m generates a unique real-time output trigger. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for upper channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 1. Therefore, these registers should be set to 0.

Table 23.108 Simultaneous Rewrite Settings for Real-Time Output Function Type 1

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.14.5 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels is available for any setting.

(2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels is available for any setting.

(3) Channel output mode for lower channels

Table 23.109 Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

| Bit Name | Setting |
|---------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: Setting is disabled (value after reset) in toggle mode. |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTRE.TAUDnTREM | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROM | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 0: Upper channel generates a real-time output trigger for channel m. |
| TAUDnTME.TAUDnTMEEm | 0: Disables modulation |

(4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

23.12.14.6 Operating Procedure for Real-Time Output Function Type 1

Table 23.110 Operating Procedure for Real-Time Output Function Type 1

| | Operation | TAUDn Status |
|-------------------------|--|--|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in Table 23.105, Contents of the TAUDnCMORm register for the Upper Channel of Real-Time Output Function Type 1 , and Table 23.106, Contents of the TAUDnCMURm register for Upper Channel of Real-Time Output Function Type 1 . | Channel operation is stopped. |
| | Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in Section 23.12.14.5, Register Settings for Lower Channels . | |
| | Set the value of TAUDnCDRm register (only channels with TAUDnTRC.TAUDnTRCm = 1) | |
| | Set channel output mode by setting the control bits as described in Table 23.107, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output . | |
| Start Operation | Set channel output mode by setting the control bits as described in Table 23.109, Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output . | |
| | Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | [Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm occurs. |
| | TAUDnCDRm and TAUDnTRO.TAUDnTROm can be changed at any time. The TAUDnCNTm register can be read at any time. | TAUDnCNTm counts down. When the counter reaches 0000 _H : <ul style="list-style-type: none"> The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. INTTAUDnIm is generated. TAUDnTTOUTm outputs the current value of the real-time output bit TAUDnTRO.TAUDnTROm. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. Both TAUDnCNTm and TAUDnTTOUTm retain their current values. |



23.12.14.7 Specific Timing Diagrams

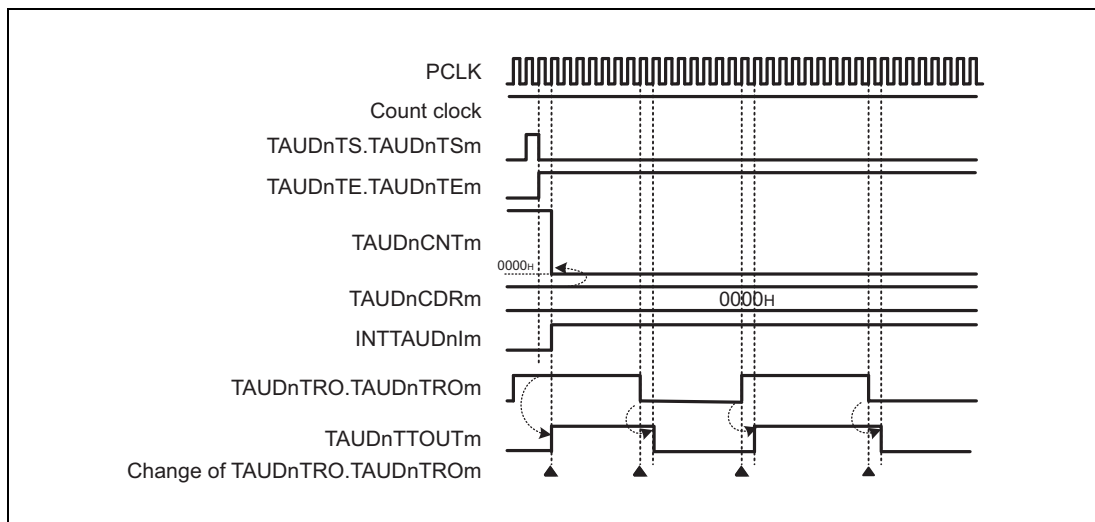


Figure 23.77 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1

23.12.15 Real-Time Output Function Type 2

23.12.15.1 Overview

Summary

This function outputs the value of TAUDnTRO.TAUDnTROm bit from TAUDnTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated when a valid TAUDnTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDnTTOUTm control of the other channels
- The operating mode for the upper channel should be set to capture mode. (See **Table 23.111, Contents of the TAUDnCMORm register for the Upper Channel of Real-Time Output Function Type 2.**)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 23.7, Channel Output Modes.**)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREM = 1).

Functional description

The counter for upper channels is started by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This sets TAUDnTE.TAUDnTEM to 1, enabling count operation. The counter starts to count up.

When a valid TAUDnTTINm input edge is generated on one of upper channels, an interrupt occurs and TAUDnTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREM = 1).

The TAUDnTTOUTm signal changes only when an interrupt is generated, and when TAUDnTTOUTm value is different to the current value of TAUDnTRO.TAUDnTROm during the occurrence of the interrupt.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREM = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREM = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.

- If the $TAUDnCMORm.TAUDnMD0$ bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 23.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

23.12.15.2 Block Diagram and General Timing Diagram

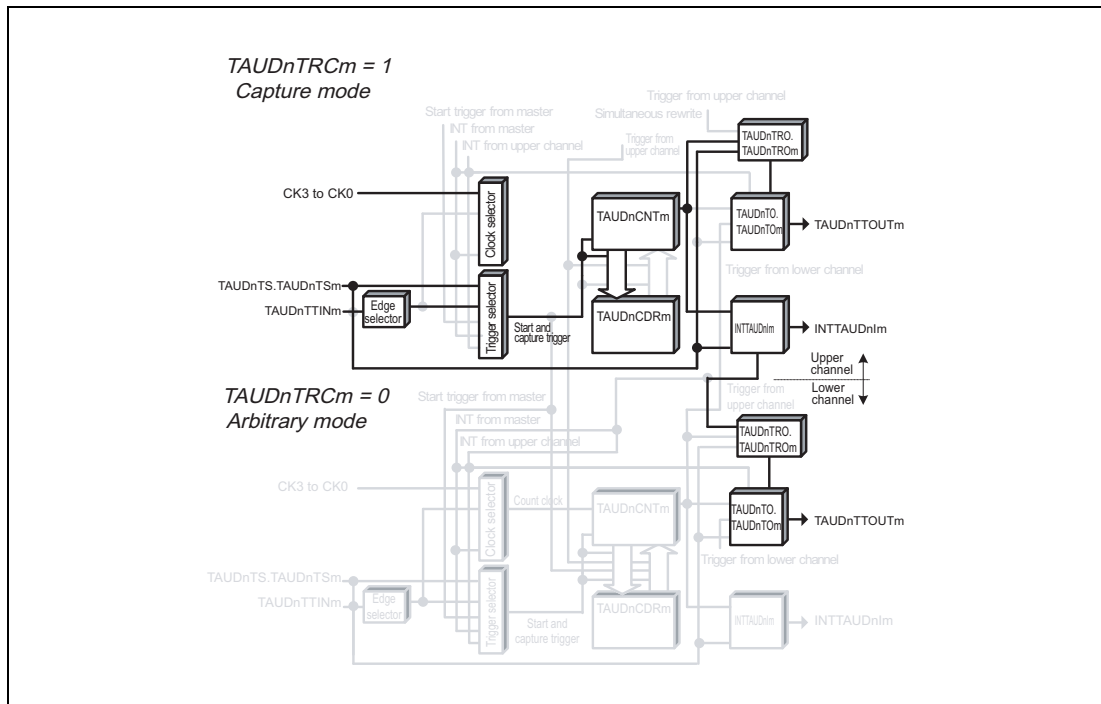


Figure 23.78 Block Diagram of Real-Time Output Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm not generated at the beginning of operation. ($TAUDnCMORm.TAUDnMD0 = 0$)

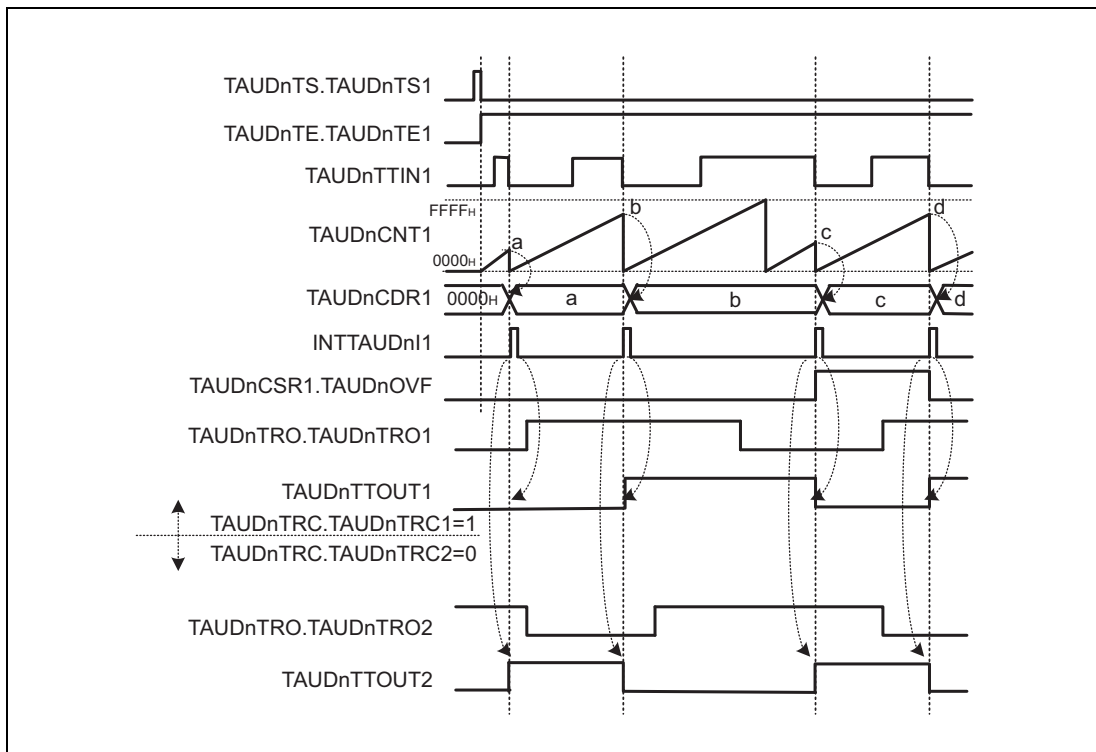


Figure 23.79 General Timing Diagram of Real-Time Output Function Type 2

23.12.15.3 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.111 Contents of the TAUDnCMORm register for the Upper Channel of Real-Time Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid edge of the TAUDnTTINm input signal is used as an external start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0010: Capture mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm for upper channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.112 Contents of the TAUDnCMURm register for the Upper Channel of Real-Time Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited |

(3) Channel output mode for upper channels**Table 23.113 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: Setting is disabled (value after reset) in toggle mode. |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROm | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 1: Channel m generates a unique real-time output trigger. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for upper channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 2. Therefore, these registers should be set to 0.

Table 23.114 Simultaneous Rewrite Settings for Real-Time Output Function Type 2

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0 |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.12.15.4 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

(2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

(3) Channel output mode for lower channels

Table 23.115 Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode. |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: Setting is disabled (value after reset) in toggle mode. |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation. |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROm | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 0: Upper channel generates a real-time output trigger for channel m. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

23.12.15.5 Operating Procedure for Real-Time Output Function Type 2

Table 23.116 Operating Procedure for Real-Time Output Function Type 2

| | Operation | TAUDn Status |
|-------------------------|--|--|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in Table 23.111, Contents of the TAUDnCMORm register for the Upper Channel of Real-Time Output Function Type 2 , and Table 23.112, Contents of the TAUDnCMURm register for the Upper Channel of Real-Time Output Function Type 2 . | Channel operation is stopped. |
| | Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in Section 23.12.15.4, Register Settings for Lower Channels . | |
| | The TAUDnCDRm register functions as a capture register (the channel of TAUDnTRC.TAUDnTRCm = 1). | |
| | Set channel output mode by setting the control bits as described in Table 23.113, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output . | |
| | Set channel output mode by setting the control bits as described in Table 23.115, Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output . | |
| Restart | Start Operation | Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. |
| | During Operation | [Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000 _H . If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm occurs. |
| Stop Operation | During Operation | TAUDnTRO.TAUDnTROm can be changed at any time. TAUDnCNTm starts to count up from 0000 _H . When a valid TAUDnTTINm input edge is detected: <ul style="list-style-type: none"> • Capture the value of TAUDnCDRm in TAUDnCNTm to clear the counter to 0000_H. • INTTAUDnIm is generated. • When TAUDnTTINm input enable edge is detected, the TAUDnCSRm.TAUDnOVF bit is set to 1 if the detection is after an overflow, and is set to 0 if the detection is before an overflow. Afterwards, this procedure is repeated. |
| | Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm, TAUDnCSRm.TAUDnOVF, and TAUDnTTOUTm retain their current values. |

23.12.15.6 Specific Timing Diagrams

(1) Operation start and stop

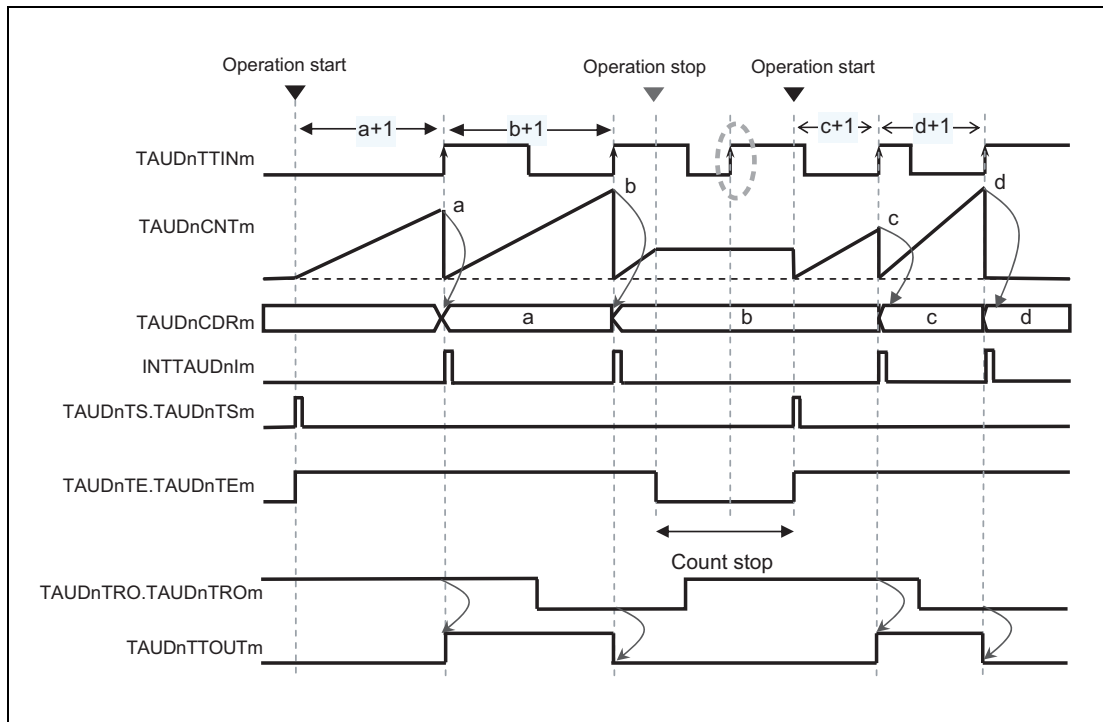


Figure 23.80 Operation Start and Stop (TAUDnCMORM.TAUDnMD0 = 0)

- When TAUDnTS.TAUDnTSm is set to 1, the counter starts counting up.
- When a valid input edge is detected, the current value of the counter is written to the data register (TAUDnCDRm) and an interrupt is generated.
- TAUDnTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROM) and the counter resets and starts to count up again.
- The TAUDnTTOUTm signal only changes when an interrupt is generated, and then only when its value is different to current value of TAUDnTRO.TAUDnTROM at the moment that the interrupt is generated.
- If the counter is stopped (TAUDnTE.TAUDnTEm is set to 0), valid input edges are ignored and no interrupt is generated.

23.12.16 Simultaneous Rewrite Trigger Generation Function Type 1

23.12.16.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 23.117, Contents of the TAUDnCMORm register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1.**)
- For the operating mode that can be set for lower channels, see **Table 23.42, Channel Functions and the Methods They Use for Simultaneous Rewrite.**
- TAUDnTTOUTm is not used for any channel in this function.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of the data register buffer for upper channels (TAUDnCDRm buf) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value. The counter for lower channels start to count according to the selected operating mode.

Once the counter reaches 0000_H, an interrupt occurs on the channel. The current value of the corresponding TAUDnCDRm buffer is loaded into TAUDnCNTm to continue operation subsequently.

If the channel where an interrupt occurs is specified as a trigger channel for simultaneous rewrite (TAUDnRDC.TAUDnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm bit should be set to 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 23.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

23.12.16.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle \times (TAUDnCDRm + 1)

To control simultaneous rewrite, the following condition should be satisfied:

[PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) \times number of interrupts] - 1

[Triangle PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) \times 2 \times number of interrupts] - 1

That is, the ratio of TAUDnCDRm + 1 and TAUDnCDRm_master + 1 should be an integer. This integer corresponds to the number of interrupts.

For triangle PWM, remember that the cycle doubles.

23.12.16.3 Block Diagram and General Timing Diagram

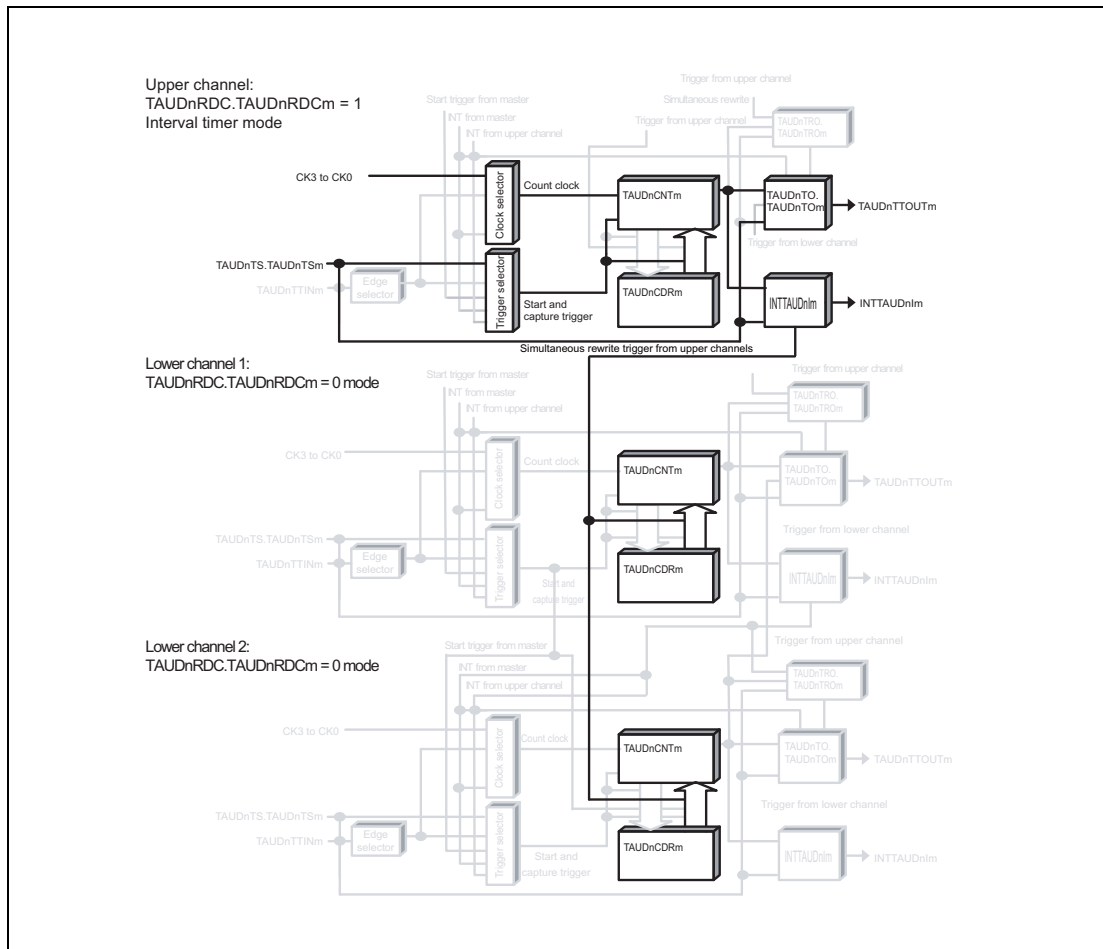


Figure 23.81 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

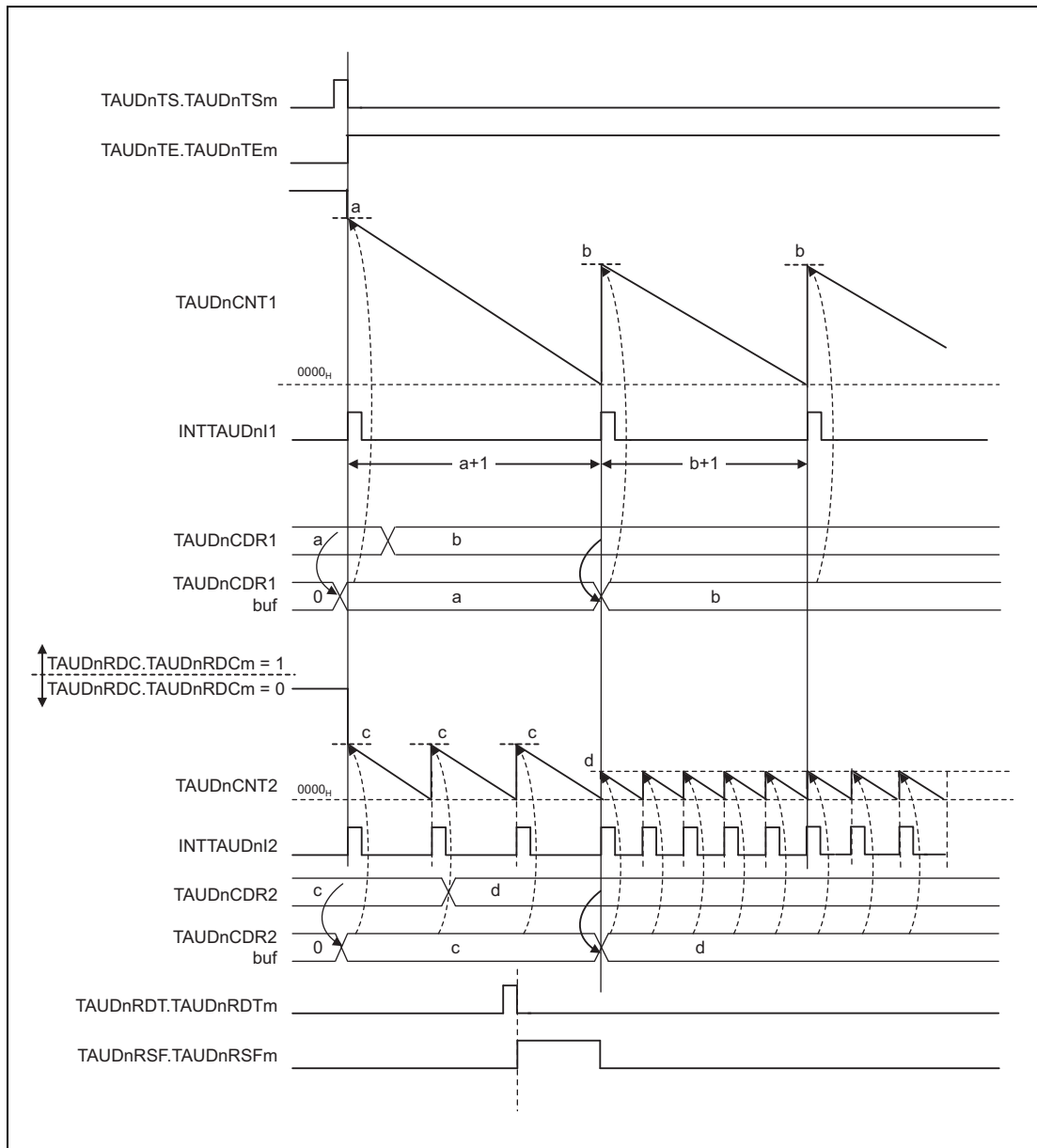


Figure 23.82 General Timing Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

23.12.16.4 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.117 Contents of the TAUDnCMORm register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Unused. Set to 0. |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm for upper channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.118 Contents of the TAUDnCMURm register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for upper channels

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous rewrite for upper channels**Table 23.119 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 1: Selects one of upper channels as simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Loads a simultaneous rewrite control signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel. |

23.12.16.5 Register Settings for Lower Channels**(1) TAUDnCMORm for lower channels**

TAUDnCMORm register for lower channels must follow the TAUDnCMORm register settings in the operating mode which can be set. (See **Table 23.42, Channel Functions and the Methods They Use for Simultaneous Rewrite.**)

(2) TAUDnCMURm for lower channels

TAUDnCMURm register for lower channels must follow the TAUDnCMURm register settings in the operating mode which can be set. (See **Table 23.42, Channel Functions and the Methods They Use for Simultaneous Rewrite.**)

(3) Channel output mode for lower channels

Output can be made according to the operating mode setting (master/slave) for lower channels. (For the functions for Simultaneous Rewrite Trigger Generation Function Type 1, see **Table 23.42, Channel Functions and the Methods They Use for Simultaneous Rewrite.**)

(4) Simultaneous rewrite for lower channels**Table 23.120 Simultaneous Rewrite Settings for Lower Channels in Simultaneous Rewrite Trigger Generation Function Type 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 1: Selects one of upper channels as simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Follow the setting of the TAUDnRDM.TAUDnRDMm in the configurable operating mode. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.12.16.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 23.121 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

| | Operation | TAUDn Status |
|-------------------------|---|--|
| Initial Channel Setting | Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in Table 23.117, Contents of the TAUDnCMORm register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1 , and Table 23.118, Contents of the TAUDnCMURm register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1 . | Channel operation is stopped. |
| | Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in Section 23.12.16.5, Register Settings for Lower Channels . | |
| | Set the value of TAUDnCDRm register. | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSM to 1. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm occurs. |
| | During Operation TAUDnRDT.TAUDnRDTm and TAUDnCDR.TAUDnCDRm are changeable. TAUDnRSF.TAUDnRSFm can be always read. | TAUDnCNTm counts down. When the counter reaches 0000H: <ul style="list-style-type: none"> The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. INTTAUDnIm is generated. If INTAUDnIm occurs on the channel where TAUDnRDC.TAUDnRDCm is set to 1, simultaneous rewrite is controlled. Afterwards, this procedure is repeated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value. |

23.13 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the timer array unit D. For a general overview of synchronous channel operation, see **Section 23.2, Overview**

This section describes functions that generate PWM signals at regular intervals.

23.13.1 PWM Output Function

23.13.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty cycle of the TAUDnTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 23.122, Contents of the TAUDnCMORm register for the Master Channel of the PWM Output Function.**)
- The operating mode for the slave channels should be set to one count mode. (See **Table 23.125, Contents of the TAUDnCMORm register for the Slave Channel of the PWM Output Function.**)
- TAUDnTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channels should be set to Synchronous Channel Output Mode 1. (See **Section 23.7, Channel Output Modes.**)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm, and the counter starts counting down from the TAUDnCDRm value. If an INTTAUDnIm occurs on the master channel and TAUDnTTOUTm (slave) is set/reset, PWM output is made.

- Master channel:
When the master channel counter reaches 0000_H and the pulse cycle time has passed, INTTAUDnIm occurs. The counter loads TAUDnCDRm value into TAUDnCNTm and counts down.

- Slave channels:

When INTTAUDnIm occurs on the master channel, the counter operation of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts counting down from the TAUDnCDRm value. TAUDnTTOUTm signal is set to the active level.

When the counter reaches to 0000_H (duty time has elapsed), INTTAUDnIm occurs and a TAUDnTTOUTm signal is set to an inactive level. The counter is reset to FFFF_H and waits for the next INTTAUDnIm (start of the next pulse cycle) of the master channel.

Counter operation can be stopped by setting the TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 23.6, Simultaneous Rewrite**.

23.13.1.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUDnCDRm (slave) / (TAUDnCDRm (master) + 1) × 100

– Duty cycle = 0%

TAUDnCDRm (slave) = 0000_H

– Duty cycle = 100%

TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

23.13.1.3 Block Diagram and General Timing Diagram

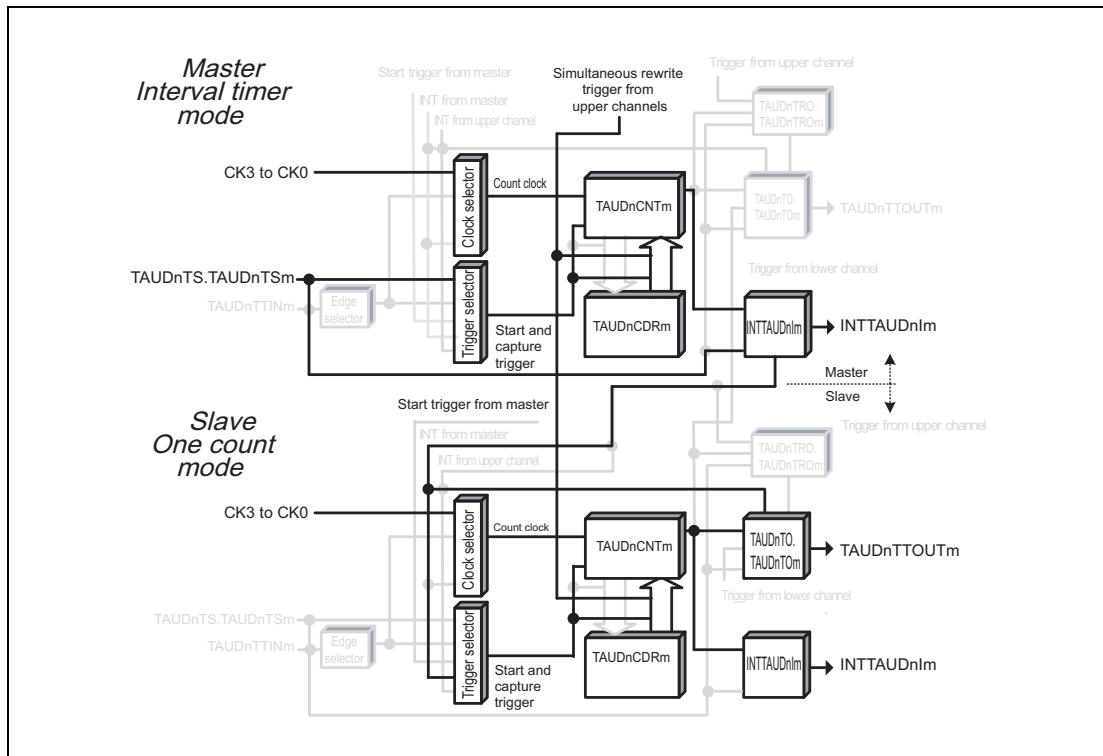


Figure 23.83 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channels: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

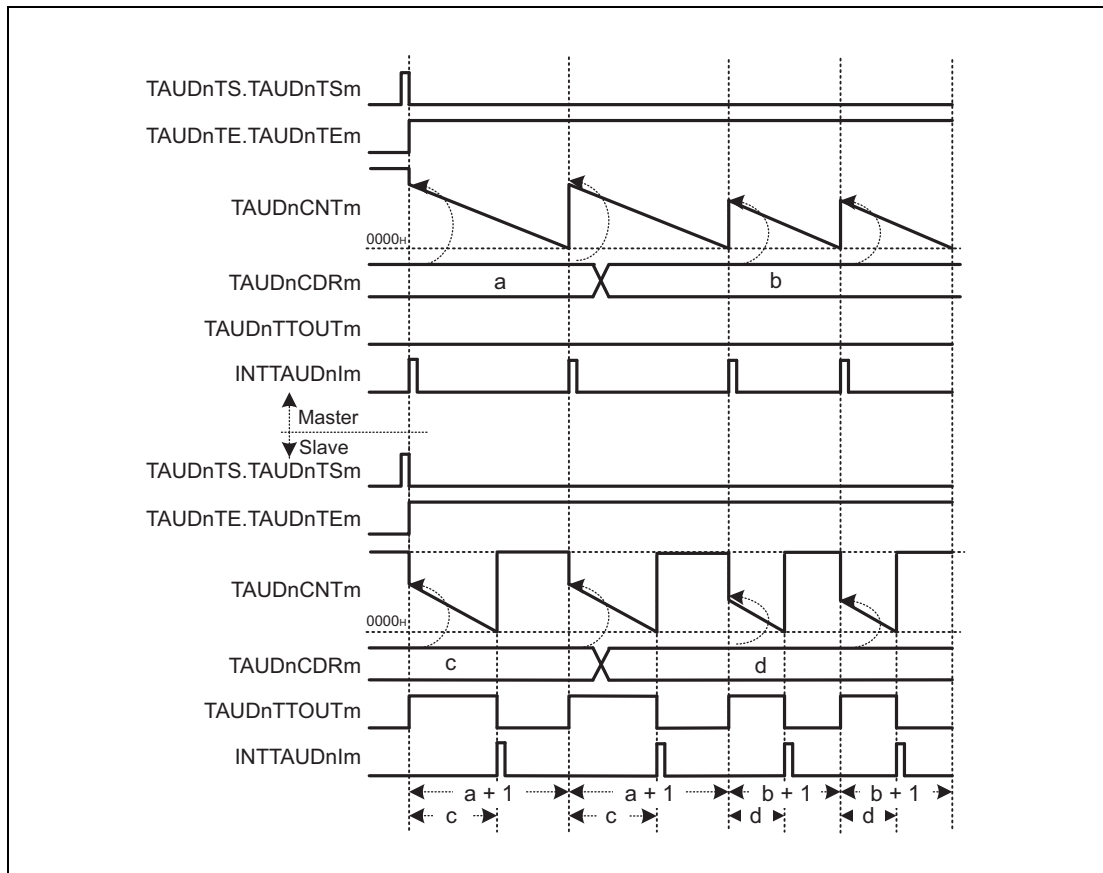


Figure 23.84 General Timing Diagram of PWM Output Function

NOTE

- The interval between the channel starting to count and an interrupt being generated is the value of corresponding TAUDnCDRm + 1.
- The TAUDTTOUTm signal from slave channel rises 1 cycle of the counter clock after the rising edge of INTTAUDnIm from the master channel.

23.13.1.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.122 Contents of the TAUDnCMORm register for the Master Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm for the master channel

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.123 Contents of the TAUDnCMURm register for the Master Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for the master channel

The channel output mode is not used with this function. However, this mode can be used with another function or in independent channel output mode controlled by software.

(4) Simultaneous rewrite for the master channel

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.124 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires a channel higher than the master channel that operates with the **Section 23.12.16, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions:

- Simultaneous rewrite trigger output function type 1 setting channel:
TAUDnRDC.TAUDnRDCm=1, TAUDnRDS.TAUDnRDSm=1
TAUDnCDRm settings for this channel are as follows:
= ((TAUDnCDRm setting for the master channel targeted for simultaneous rewrite + 1) × interrupt count) - 1
- Master channels: TAUDnRDC.TAUDnRDCm=0, TAUDnRDS.TAUDnRDSm=1
- Slave channels: TAUDnRDC.TAUDnRDCm=0, TAUDnRDS.TAUDnRDSm=1

If TAUDnCDRm (slave) setting > TAUDnCDRm (master) setting + 1, the duty value (which exceeds 100%) is aggregated to be 100% output.

23.13.1.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKs[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.125 Contents of the TAUDnCMORm register for the Slave Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKs[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKs[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 100: INTTAUDnIm of master channel is a start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Start trigger during operation is valid. |

(2) TAUDnCMURm for slave channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.126 Contents of the TAUDnCMURm register for the Slave Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channels**Table 23.127 Control Bit Settings in Synchronous Channel Output Mode 1**

| Bit Name | Setting |
|--|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel operation |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm TAUDnTDL.TAUDnTDLm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm TAUDnTRC.TAUDnTRCm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for slave channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.128 Simultaneous Rewrite Settings for Slave Channels of PWM Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.1.6 Operating Procedure for PWM Output Function

Table 23.129 Operating Procedure for PWM Output Function

| | Operation | TAUDn Status |
|-------------------------|---|--|
| Initial Channel Setting | Master channels: Set TAUDnCMORM and TAUDnCMURm registers and the channel output mode as described in Section 23.13.1.4, Register Settings for the Master Channel. | Channel operation is stopped. |
| | Slave channels: Set TAUDnCMORM and TAUDnCMURm registers and the channel output mode as described in Section 23.13.1.5, Register Settings for Slave Channels. | |
| | Set the value of TAUDnCDRm register of every channel. | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel and TAUDnTTOUTm (slave) is set. |
| | During operation TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation. | TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000 _H : <ul style="list-style-type: none"> • INTTAUDnIm (master) occurs. • TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down. • TAUDnTTOUTm (slave) is set to the active level. If TAUDnCNTm (slave) reaches 0000 _H : <ul style="list-style-type: none"> • INTTAUDnIm (slave) occurs. • TAUDnTTOUTm (slave) is set to an inactive level. Count operation of the slave channel stops. |
| Stop Operation | Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.13.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

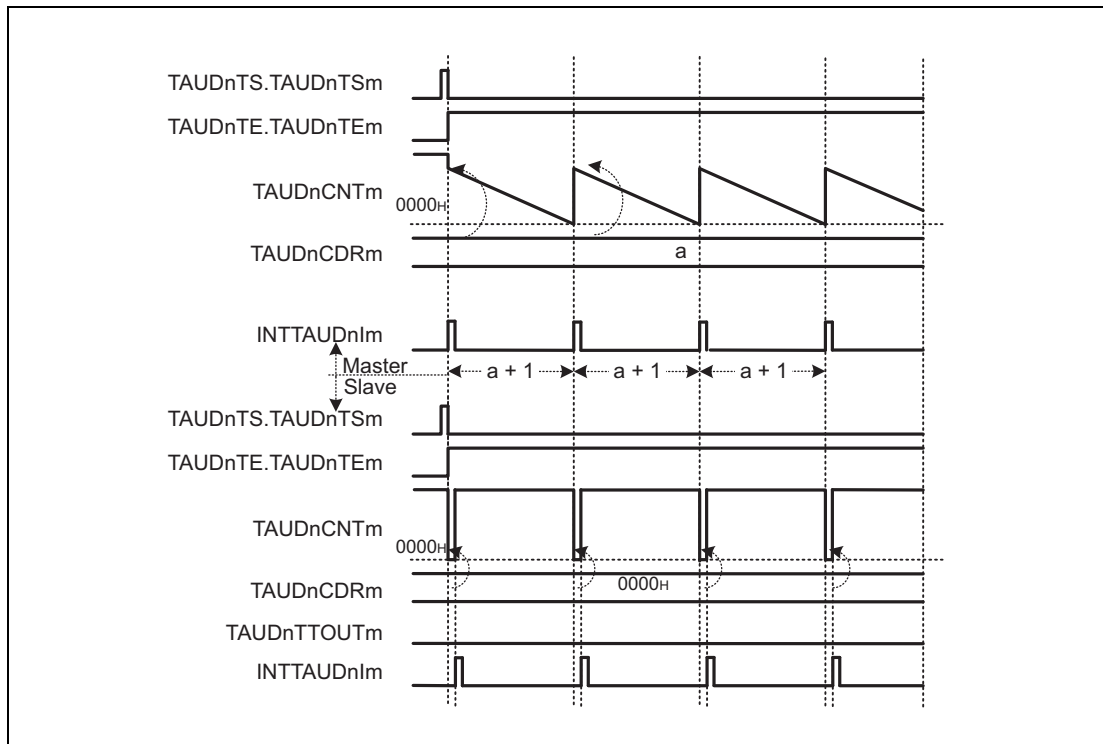


Figure 23.85 TAUDnCDRm (Slave) = 0000_H,
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000_H is loaded in to TAUDnCNTm (slave). Therefore, an interrupt (INTTAUDnIm) of the slave channel occurs simultaneously and TAUDnTTOUTm remains inactive.
- TAUDnCDRm value is loaded into TAUDnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

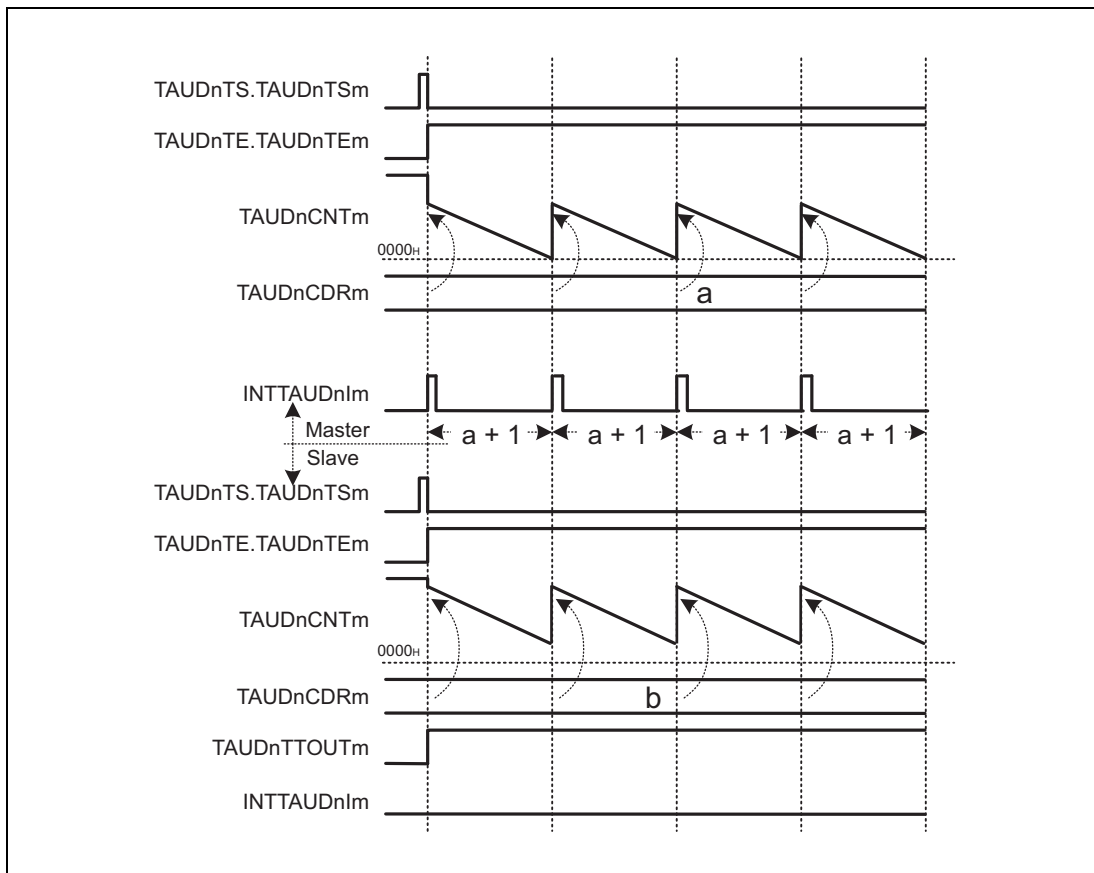


Figure 23.86 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1
 Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- If TAUDnCDRm (slave) value is greater than TAUDnCDRm (master) value, the slave channel counter does not reach 0000_H and consequently, no interrupt occurs. TAUDnTTOUTm remains active.

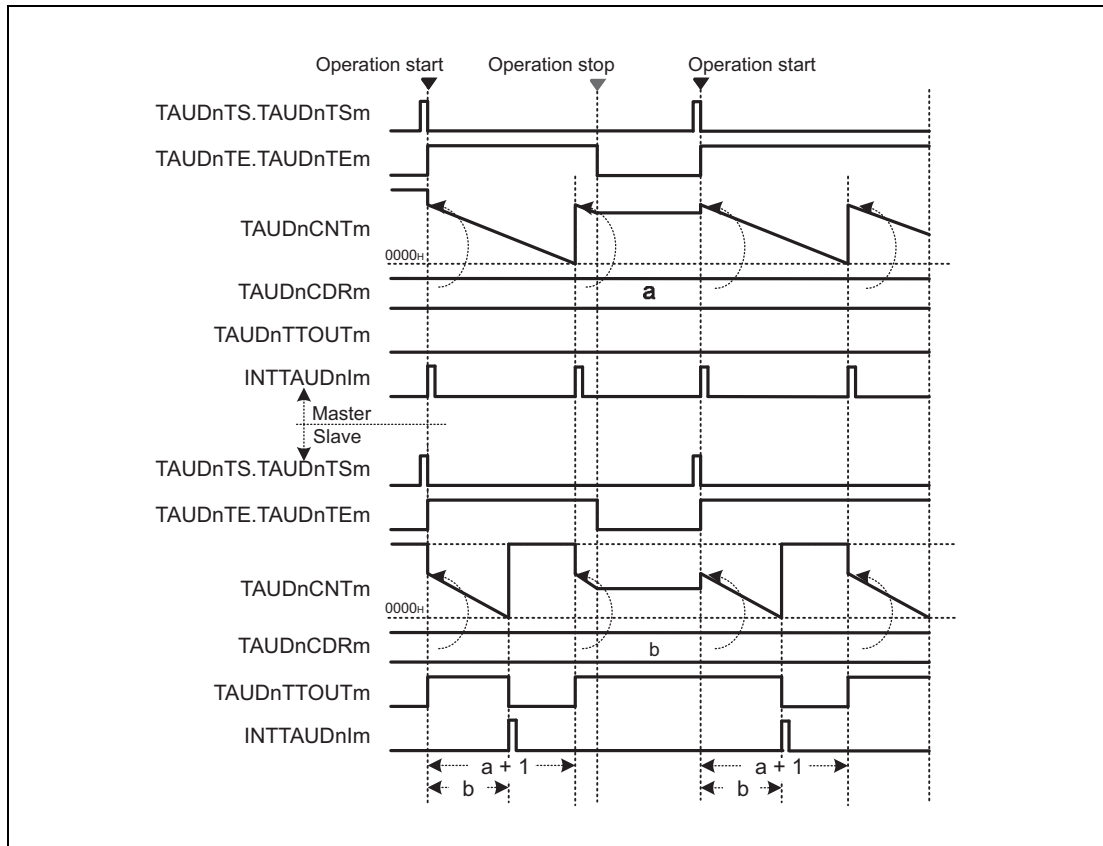
(3) Operation stop and restart

Figure 23.87 Operation Stop and Restart
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDnTTOUtm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm of master and slave channels to 1. TAUDnCDRm values of the master and slave channels are loaded to TAUDnCNTm and start to count down from these values.

23.13.2 One-Shot Pulse Output Function

23.13.2.1 Overview

Summary

This function outputs a signal pulse with a specific pulse width and delay time (both defined relative to an external input signal pulse or software trigger) by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to one-count mode. (See **Table 23.130, Contents of the TAUDnCMORm register for the Master Channel of the One-Shot Pulse Output Function.**)
- The operating mode for slave channels should be set to pulse one-count mode. (See **Table 23.133, Contents of the TAUDnCMORm register for the Slave Channel of the One-Shot Pulse Output Function.**)
- TAUDnTTOUtm is not used with the master channel of this function.
- The channel output mode for the slave channel should be set to independent channel output mode 2. (See **Section 23.7, Channel Output Modes.**)
- TAUDnTTINm (master) has to be detected while TAUDnCNTm (master) and TAUDnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUDnTTINm (slave).
- When using only the software trigger, do not select TAUDnTTINm with the alternative function of the pins.

Functional description

The counters are enabled by setting the channel trigger bits of the master channel and slave channel (TAUDnTS.TAUDnTSM) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
When the next valid TAUDnTTINm input edge or software trigger (when TAUDnTE.TAUDnTEm = 1, TAUDnTS.TAUDnTSM = 1 (m: master channel number)) is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm. The counter starts to count down from this value. If TAUDnCMORm.TAUDnMD0 = 0, a trigger (TAUDnTTINm) which is detected within the delay time is ignored.
When the counter of master channel reaches 0000_H, INTTAUDnIm is generated. The counter is reset to FFFF_H and waits for the next valid TAUDnTTINm input edge or software trigger (when TAUDnTE.TAUDnTEm = 1, TAUDnTS.TAUDnTSM = 1 (m: master channel number)).
- Slave channels:
INTTAUDnIm generated on master channel triggers the counter operation of slave channel. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave). The counter starts counting down from this value. An interrupt occurs and the TAUDnTTOUtm signal is set.
When the counter reaches 0001_H, INTTAUDnIm is generated and TAUDnTTOUtm signal is reset. The counter stops at 0000_H and waits for the next INTTAUDnIm of master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnCMORm.TAUDnMD0 of master channel is set to 0, TAUDnTTINm input edges detected during counting are ignored.
- Simultaneous rewrite can be used with this function. See **Section 23.6, Simultaneous Rewrite**.

23.13.2.2 Equations

Delay from trigger input to pulse output = (TAUDnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

23.13.2.3 Block Diagram and General Timing Diagram

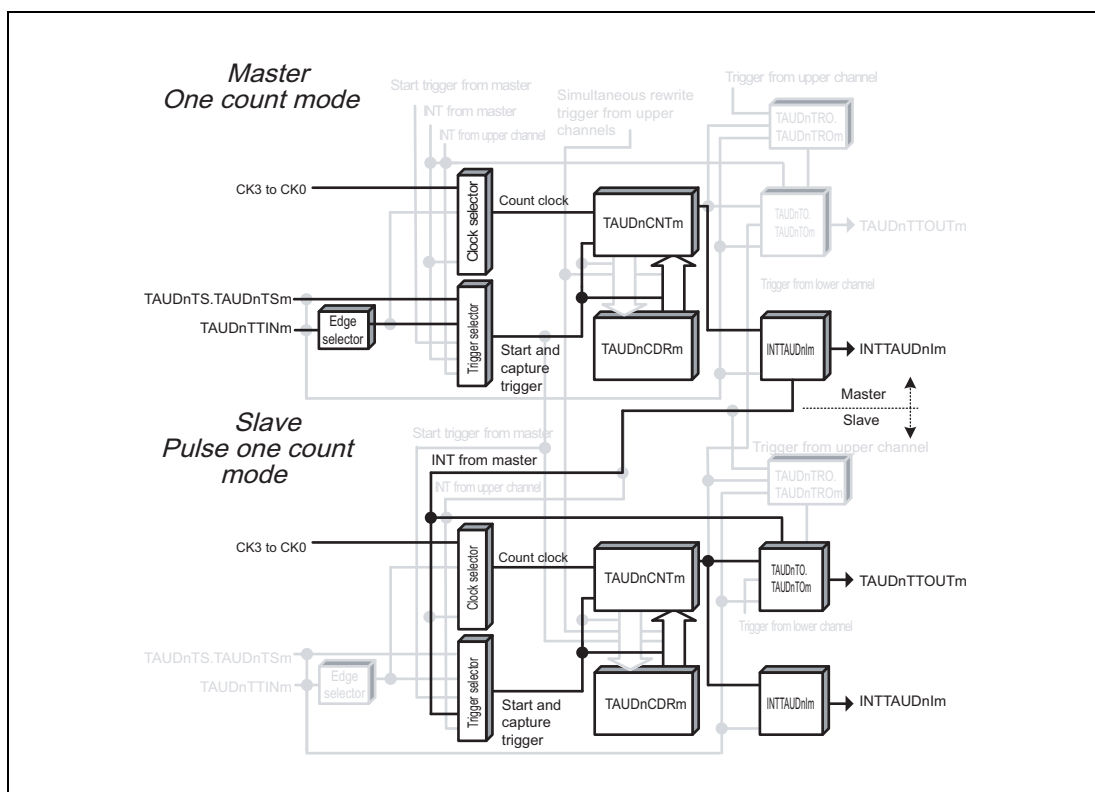


Figure 23.88 Block Diagram of One-Shot Pulse Output Function

The following settings apply to **Figure 23.89, General Timing Diagram of One-Shot Pulse Output Function (External Input Signal)** and **Figure 23.90, General Timing Diagram of One-Shot Pulse Output Function (Software Trigger)**.

- Start trigger detection is disabled during counting (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

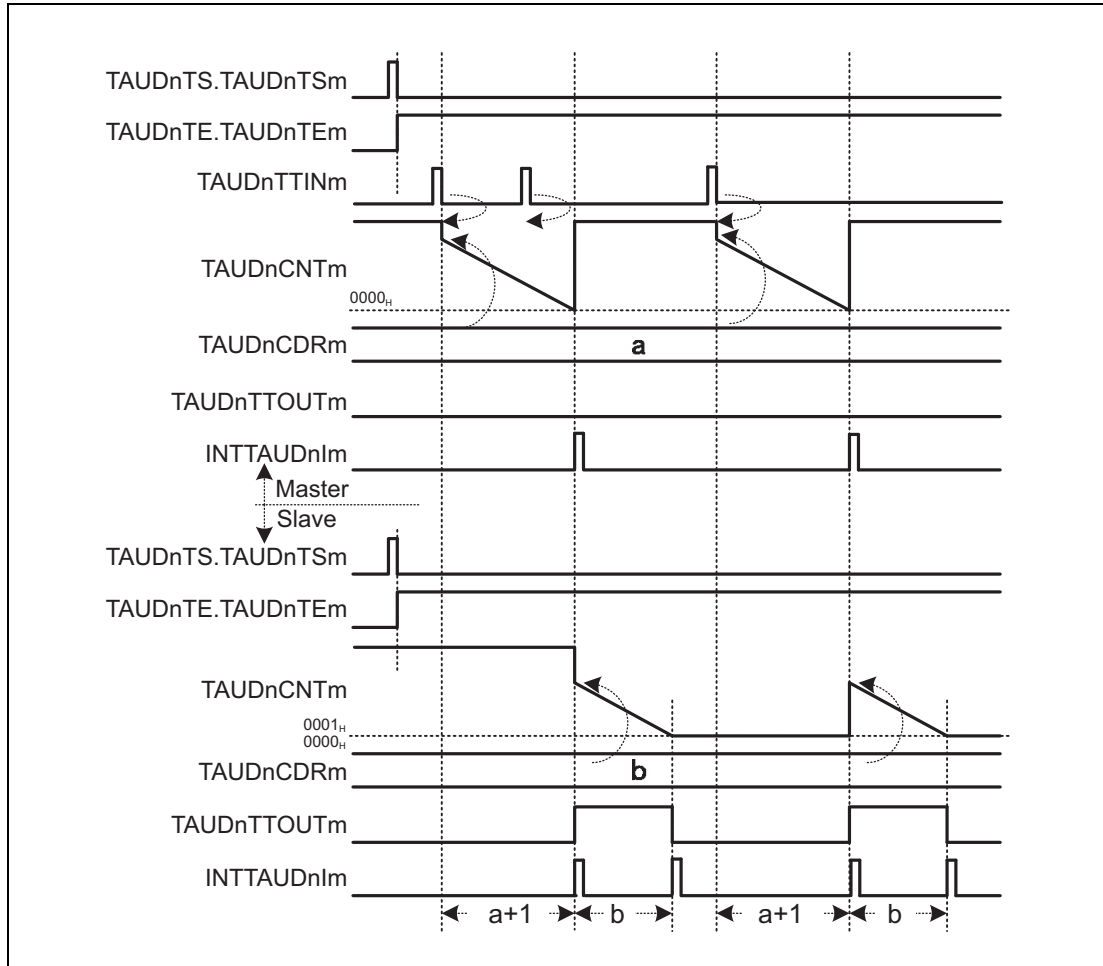


Figure 23.89 General Timing Diagram of One-Shot Pulse Output Function (External Input Signal)

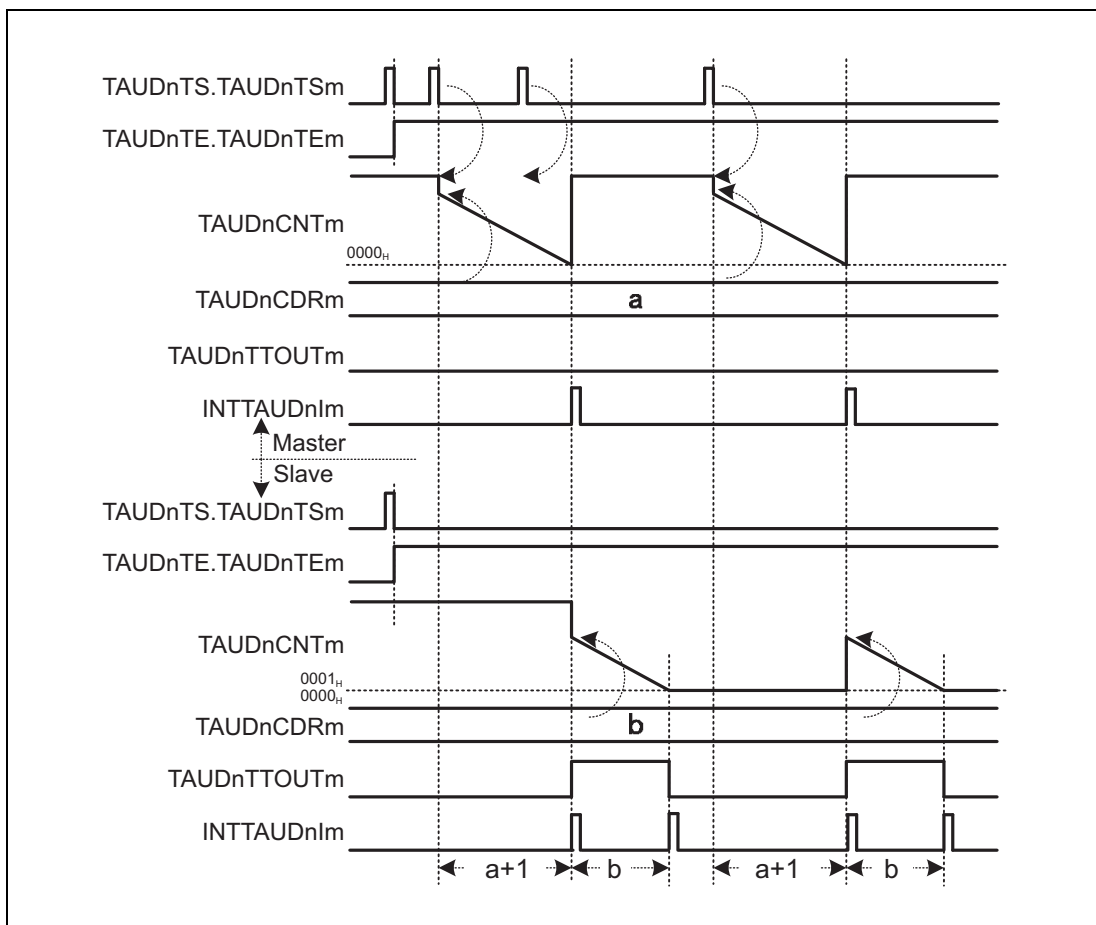


Figure 23.90 General Timing Diagram of One-Shot Pulse Output Function (Software Trigger)

23.13.2.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.130 Contents of the TAUDnCMORm register for the Master Channel of the One-Shot Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 001: Valid TAUDnTTINm input edge signal is used as the start trigger |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 0: Disables detection of start trigger during count operation. 1: Enables detection of start trigger during count operation. MD0 bit of master and slave channels should have the same value. |

(2) TAUDnCMURm for master channels

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.131 Contents of the TAUDnCMURm register for the Master Channel of the One-Shot Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited |

(3) Channel output mode for master channels

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous rewrite for master channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.132 Simultaneous Rewrite Settings for Master Channels of One-Shot Pulse Output Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Master channel is simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.2.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.133 Contents of the TAUDnCMORm register for the Slave Channel of the One-Shot Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 100: INTTAUDnIm of master channel is a start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1010: Pulse one-count mode |
| 0 | TAUDnMD0 | 0: Disables detection of start trigger during count operation. 1: Enables start trigger detection while counting. MD0 bit of master and slave channels should have the same value. |

(2) TAUDnCMURm for slave channels

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.134 Contents of the TAUDnCMURm register for the Slave Channel of the One-Shot Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for the slave channel**Table 23.135 Control Bit Settings in Independent Channel Output Mode 2**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for slave channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.136 Simultaneous Rewrite Settings for Slave Channels of One-Shot Pulse Output Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Master channel is simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 23.137 Operating Procedure for One-Shot Pulse Output Function

| | Operation | TAUDn Status |
|-------------------------|--|--|
| Initial Channel Setting | Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.2.4, Register Settings for the Master Channel. | Channel operation is stopped. |
| | Slave channels: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in Section 23.13.2.5, Register Settings for Slave Channels. | |
| | Set the value of TAUDnCDRm register of every channel. | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSM of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUDnTTINm input. |
| During Operation | TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. | When valid TAUDnTTINm input edge is detected, TAUDnCDRm value of master channel is loaded into TAUDnCNTm to start countdown. When the counter reaches 0000H: <ul style="list-style-type: none"> • INTTAUDnIm (master) occurs. • TAUDnCNTm (master) returns to FFFFH, and awaits for the next valid TAUDnTTINm input edge. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to start countdown. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to start countdown. • INTTAUDnIm (slave) occurs. • TAUDnTTOUTm (slave) becomes active level. When TAUDnCNTm (slave) reaches 0001H: <ul style="list-style-type: none"> • INTTAUDnIm (slave) occurs. • TAUDnTTOUTm (slave) becomes inactive and count operation of the slave channel stops. |
| | TAUDnRDT.TAUDnRDTm can be changed during operation. | |
| Stop Operation | Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.13.2.7 Specific Timing Diagrams

(1) TAUDnCDRm (master) = 0000_H

The following settings apply to this diagram:

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

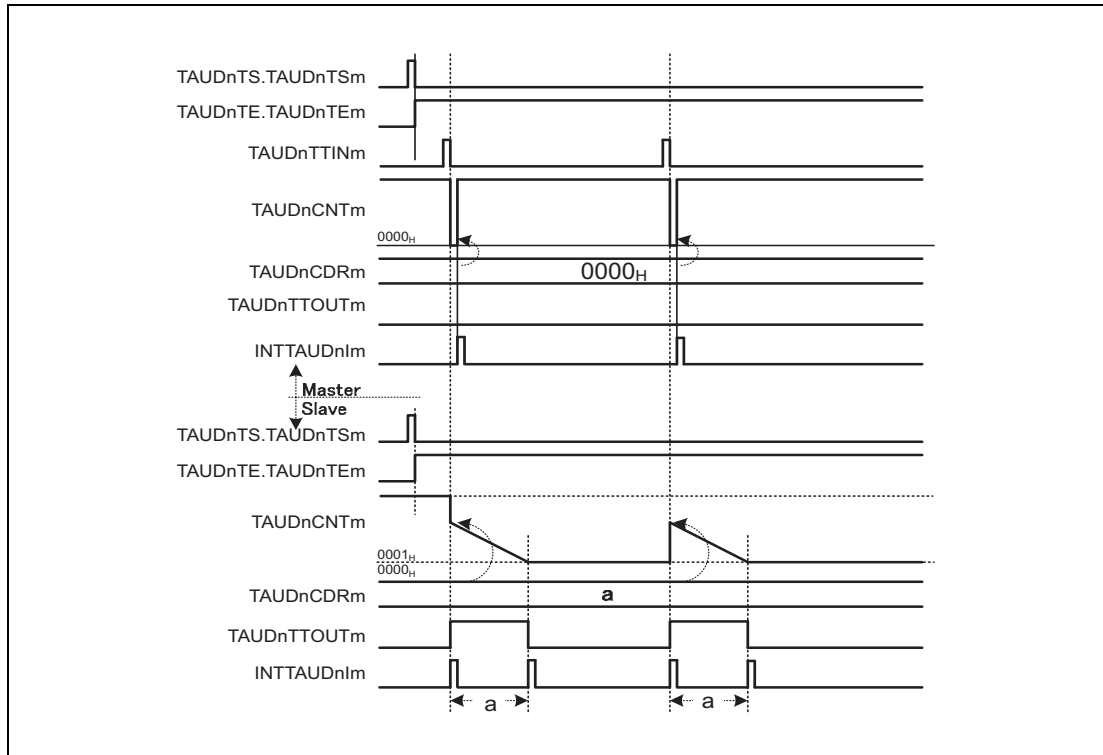


Figure 23.91 TAUDnCDRm (Master) = 0000_H

- When a valid TAUDnTTINm input edge is detected, the value 0000_H is written to TAUDnCNTm (master). The counter is set to 0000_H for one count and returns to FFFF_H. Thus the slave channel starts to count down one count clock later to TAUDnTTINm (master).

(3) TAUDnCMORm.TAUDnMD0 = 1

The following settings apply to this diagram:

- Enables start trigger detection while counting. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

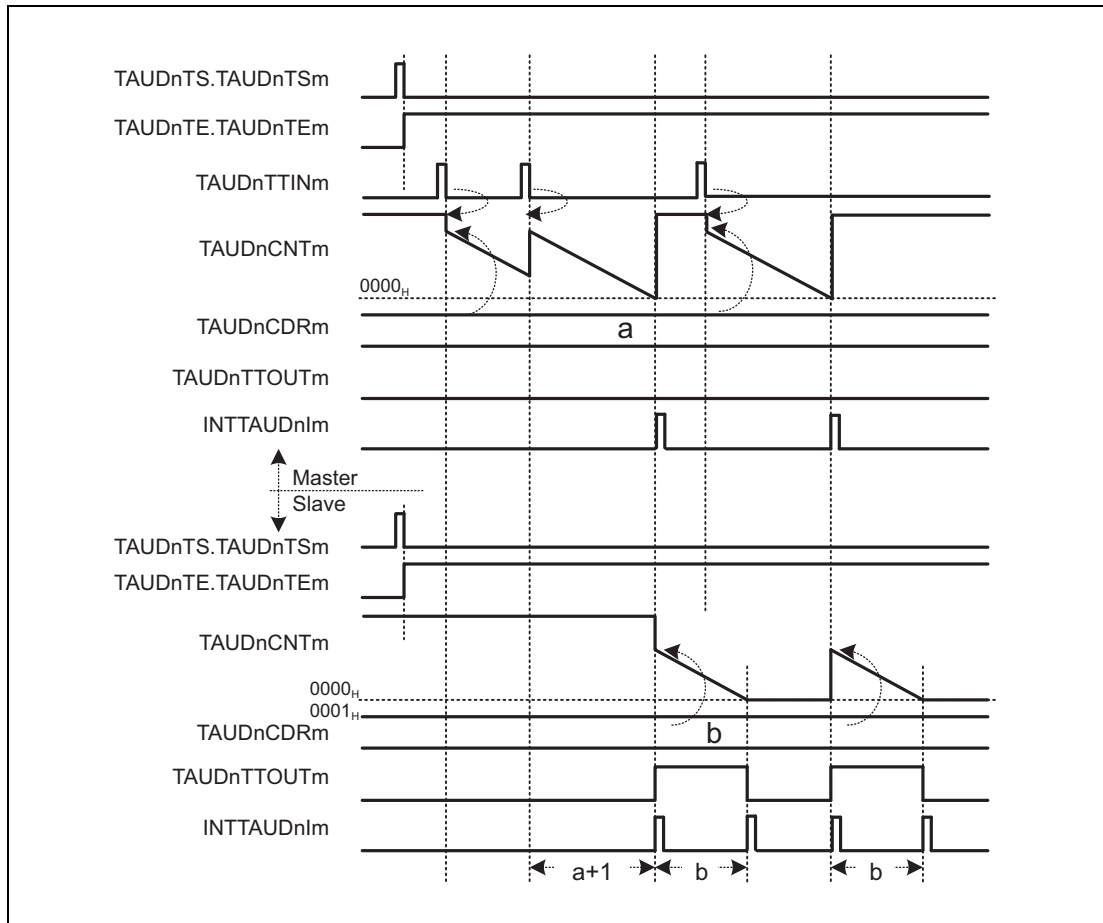


Figure 23.93 TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDnTTINm input edge is detected while the counter of the master channel counts down, TAUDnCNTm reloads the value of TAUDnCDRm. The counter restarts to count down. This means the delay is extended by the value of TAUDnCNTm at the time a valid TAUDnTTINm input edge is detected.

(4) Restarting the master channel while the slave channel is counting

The following settings apply to this diagram:

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

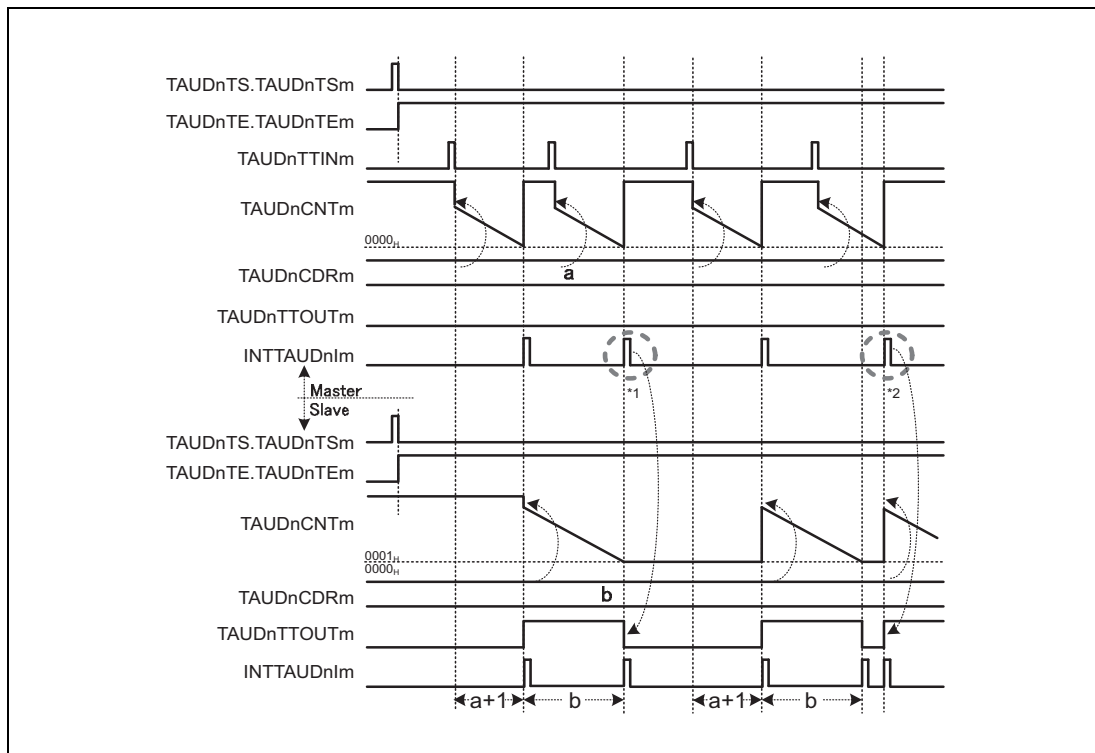


Figure 23.94 Input Interval of TAUDnTTINm ≤ Delay Time + Pulse Width + 1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001_H or exactly when 0001_H is reached*¹, the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUDnCDRm (slave) is reloaded. An interrupt is generated and TAUDnTTOUTm toggles. If TAUDnCNTm (master) has started to count down while the TAUDnCNTm (slave) is still counting*², TAUDnTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

23.13.3 Delay Pulse Output Function

23.13.3.1 Overview

Summary

This function outputs two signals. The pulse width and pulse cycle of the reference signal are defined using the master channel and slave channel 1. Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by the amount specified on slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1. The duty cycle of the delay signal is specified using slave channel 3.
- The delay amount is specified on slave channel 2.

Prerequisites

- Four channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 23.138, Contents of the TAUDnCMORm register for the Master Channel of the Delay Pulse Output Function.**)
- The operating mode for slave channels 1 and 2 should be set to one-count mode. (See **Table 23.141, Contents of the TAUDnCMORm register for the Slave Channel 1 of the Delay Pulse Output Function, Table 23.145, Contents of the TAUDnCMORm register for the Slave Channel 2 of the Delay Pulse Output Function.**)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See **Table 23.148, Contents of the TAUDnCMORm register for the Slave Channel 3 of the Delay Pulse Output Function.**)
- TAUDnTTOUTm is not used with the master channel and slave channel 2.
- The channel output mode for slave channel 1 should be set to synchronous channel output mode 1. (See **Section 23.7, Channel Output Modes.**)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See **Section 23.7, Channel Output Modes.**)

Functional description

The counters of the channel group are started by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel.
When the counter value of master channel reaches 0000_H and pulse cycle time has elapsed, INTTAUDnIm is generated. The TAUDnCDRm value is reloaded into the counter to perform count down.

- Slave channels 1 and 2:

Slave channels 1 and 2 start to count down from the current TAUDnCDRm value when detecting an interrupt from the master channel. TAUDnTTOUTm signal (slave 1) is set.

 - Slave channel 1:

When the counter of slave channel 1 reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and TAUDnTTOUTm signal is reset. The counter is reset to FFFF_H and waits for the next INTTAUDnIm of master channel.
 - Slave channel 2:

When the counter of slave channel 2 reaches 0000_H and delay time has elapsed, INTTAUDnIm is generated. The counter is reset to FFFF_H and waits for the next INTTAUDnIm of master channel.

Generating INTTAUDnIm (slave channel 2) triggers the counter of slave channel 3.
- Slave channel 3:

When slave channel 3 detects an interrupt from slave channel 2, its counter starts counting down from the current value of TAUDnCDRm. INTTAUDnIm is generated and the TAUDnTTOUTm signal (slave channel 3) is set.

When the counter of slave channel 3 reaches 0001_H, INTTAUDnIm is generated and the TAUDnTTOUTm signal is reset.

The delayed PWM pulse is output from slave channel 3.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 23.6, Simultaneous Rewrite**.

23.13.3.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUDnCDRm (slave 1)) × count clock cycle

Delay width = (TAUDnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUDnCDRm (slave 3)) × count clock cycle

However, the delay width shall be set within the following range:

$0000_{\text{H}} \leq \text{TAUDnCDRm (slave 2)} < \text{TAUDnCDRm (master)}$

NOTES

1. The output waveform of TAUDnTTOUTm (slave 3) becomes the waveform made by delaying the output waveform of TAUDnTTOUTm (slave 1) by the quantity generated by slave 2. It is impossible to make a delay longer than the pulse cycle.
2. If INTTAUDnIm of slave 2 is generated while slave 3 is counting, slave 3 restarts operation. Therefore, the output waveform of TAUDnTTOUTm (slave 3) is retained on the active level. In this case, TOUTn (Slave-CH-3) cannot output the waveform generated by delaying the basic pulse of TOUTn (Slave-CH-1).

23.13.3.3 Block Diagram and General Timing Diagram

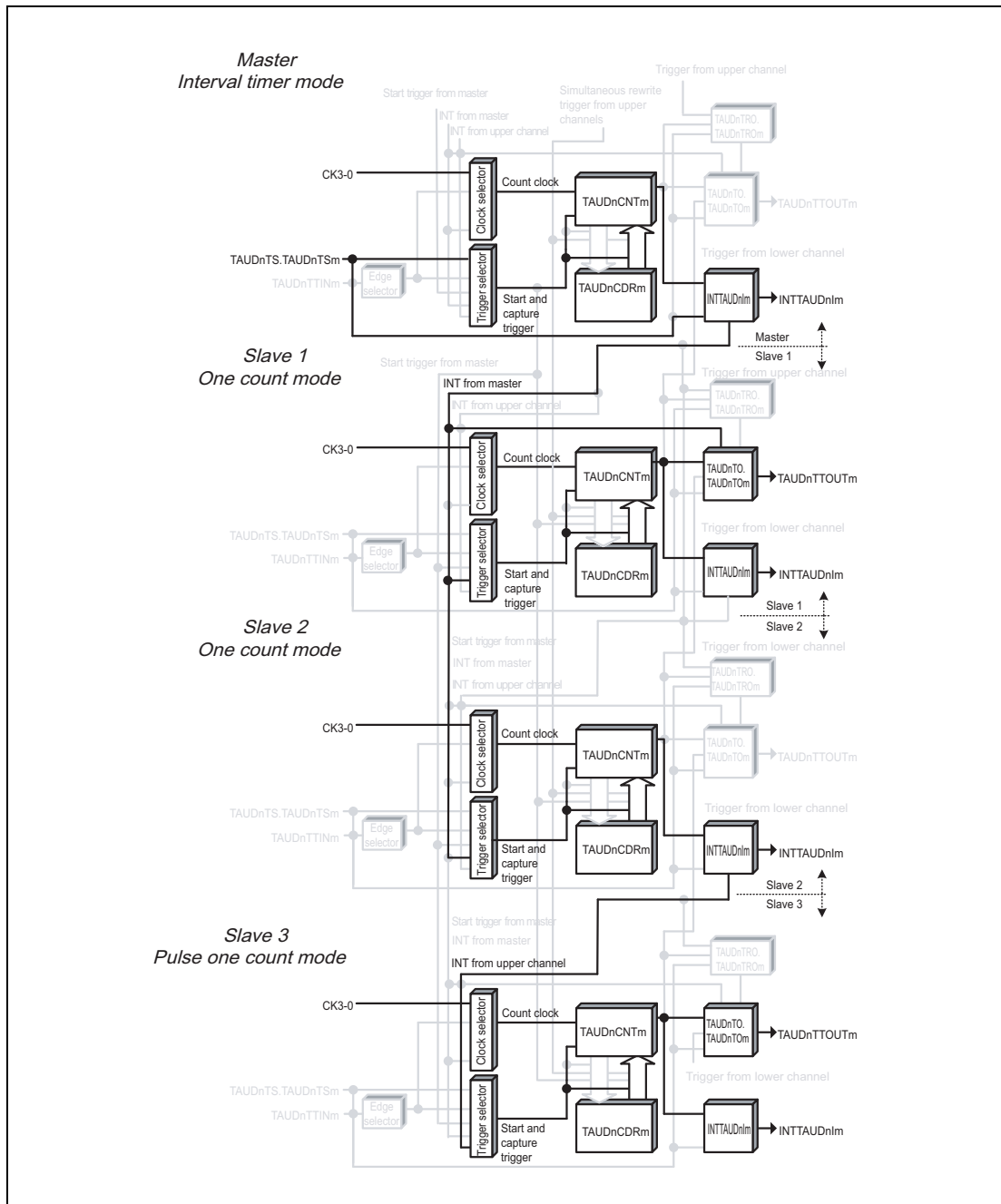


Figure 23.95 Block Diagram of Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUDnTOL.TAUDnTOLm = 0)
- Slave channel 3: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

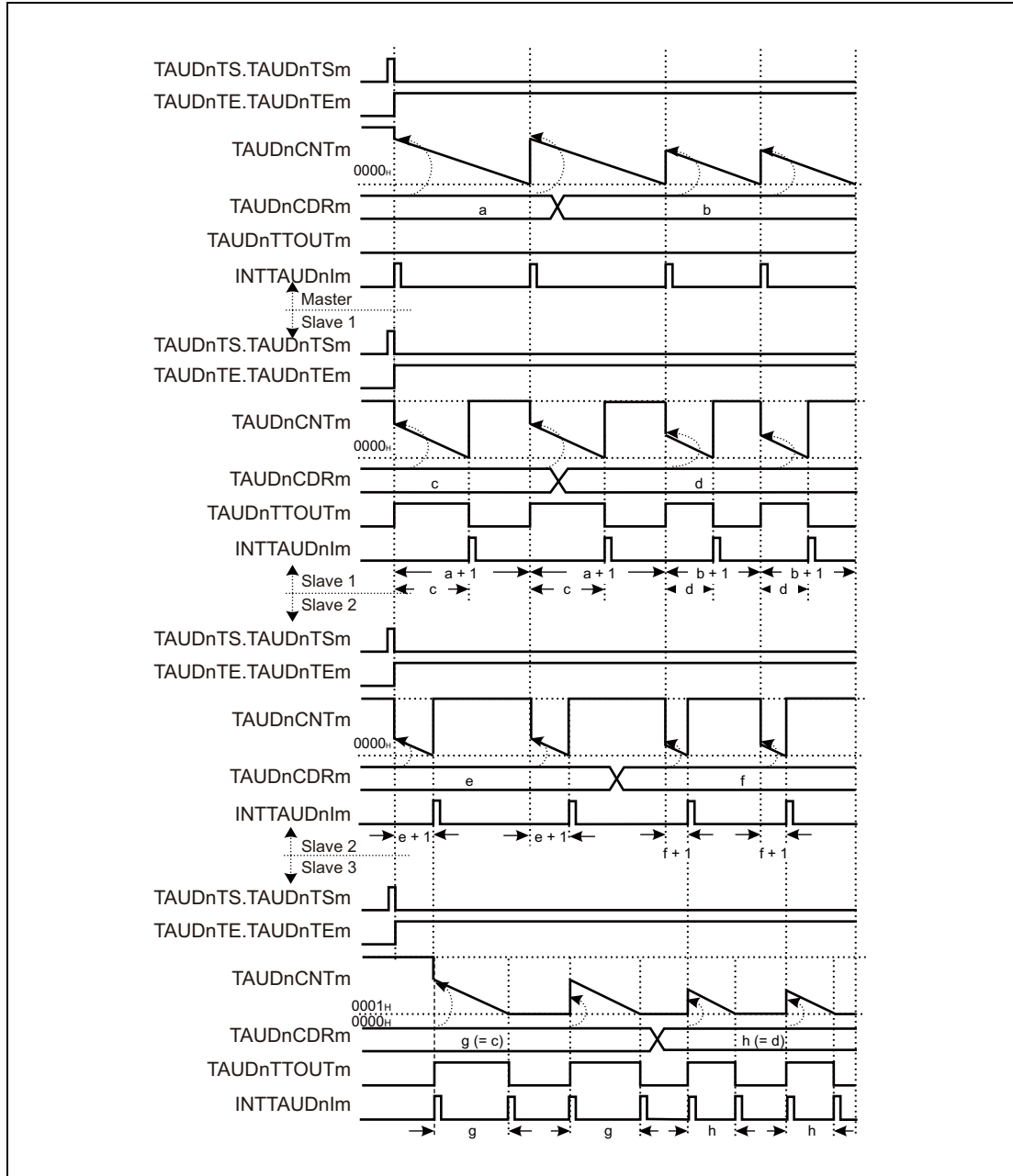


Figure 23.96 General Timing Diagram of Delay Pulse Output Function

23.13.3.4 Register Settings for the Master Channels

(1) TAUDnCMORm for the master channel

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.138 Contents of the TAUDnCMORm register for the Master Channel of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm for the master channel

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.139 Contents of the TAUDnCMURm register for the Master Channel of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used for master channels with this function.

(4) Simultaneous rewrite for the master channel

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.140 Simultaneous Rewrite Settings for the Master Channel of Delay Pulse Output Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Master channel is simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not use the channel to generate the simultaneous rewrite trigger. |

23.13.3.5 Register Settings for the Slave Channel 1

(1) TAUDnCMORm for the slave channel 1

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.141 Contents of the TAUDnCMORm register for the Slave Channel 1 of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 100: INTTAUDnIm of master channel is a start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Valid start trigger during operation |

(2) TAUDnCMURm for the slave channel 1

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.142 Contents of the TAUDnCMURm register for the Slave Channel 1 of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for the slave channel 1**Table 23.143 Control Bit Settings for the Slave Channel 1 in Synchronous Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel operation |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for the slave channel 1

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.144 Simultaneous Rewrite Settings for the Slave Channel 1 of Delay Pulse Output Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Master channel is simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.3.6 Register Settings for Slave Channel 2

(1) TAUDnCMORm for slave channel 2

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKs[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.145 Contents of the TAUDnCMORm register for the Slave Channel 2 of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKs[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKs[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 100: INTTAUDnIm of master channel is a start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Valid start trigger during operation |

(2) TAUDnCMURm for slave channel 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.146 Contents of the TAUDnCMURm register for the Slave Channel 2 of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channel 2

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous rewrite for slave channel 2

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.147 Simultaneous Rewrite Settings for Slave Channel 2 of Delay Pulse Output Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Master channel is simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.3.7 Register Settings for Slave Channel 3

(1) TAUDnCMORm for slave channel 3

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.148 Contents of the TAUDnCMORm register for the Slave Channel 3 of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 101: INTTAUDnIm of upper channel (m - 1) is a start trigger regardless of master setting. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1010: Pulse one-count mode |
| 0 | TAUDnMD0 | 1: Valid start trigger during operation |

(2) TAUDnCMURm for slave channel 3

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.149 Contents of the TAUDnCMURm register for the Slave Channel 3 of the Delay Pulse Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channel 3**Table 23.150 Control Bit Settings in Independent Channel Output Mode 2**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for slave channel 3

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.151 Simultaneous Rewrite Settings for Slave channel 3 of Delay Pulse Output Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Master channel is simultaneous rewrite control channel. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not use the channel to generate the simultaneous rewrite trigger. |

23.13.3.8 Operating Procedure for Delay Pulse Output Function

Table 23.152 Operating Procedure for Delay Pulse Output Function (1/2)

| | Operation | TAUDn Status |
|-------------------------|---|-------------------------------|
| Initial Channel Setting | Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.3.4, Register Settings for the Master Channels. | Channel operation is stopped. |
| | Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.3.5, Register Settings for the Slave Channel 1. | |
| | Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.3.6, Register Settings for Slave Channel 2. | |
| | Slave channel 3: set the TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.3.7, Register Settings for Slave Channel 3. | |
| | Set the value of TAUDnCDRm register of every channel. | |

Table 23.152 Operating Procedure for Delay Pulse Output Function (2/2)

| | Operation | TAUDn Status |
|----------------|---|--|
| Restart | Start Operation Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master channel and slave channels 1 and 2 start. INTTAUDnIm is generated on the master channel and TAUDnTTOUTm (slave channel 1) is set. |
| | During Operation TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation. | TAUDnCNTm of master channel and slave channels 1 and 2 load TAUDnCDRm value and count down. When the counter of master channel reaches 0000H: <ul style="list-style-type: none"> • INTTAUDnIm (master) occurs. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1/2) to start countdown. • TAUDnTTOUTm (slave 1) is set. When TAUDnCNTm (slave 1) reaches 0000H: <ul style="list-style-type: none"> • INTTAUDnIm (slave 1) occurs. • TAUDnTTOUTm (slave 1) is reset. When TAUDnCNTm (slave 2) reaches 0000H: <ul style="list-style-type: none"> • INTTAUDnIm (slave 2) occurs. • INTTAUDnIm (slave 3) occurs. • TAUDnTTOUTm (slave 3) is set. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 3) to start a countdown operation. When TAUDnCNTm (slave 3) reaches 0001H: <ul style="list-style-type: none"> • INTTAUDnIm (slave 3) occurs. • TAUDnTTOUTm (slave 3) is reset. |
| Stop Operation | Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.13.3.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to **Figure 23.97**:

- TAUDnCDRm(master) = 000A_H
- TAUDnCDRm(slave 1) = 000B_H
- TAUDnCDRm(slave 2) = 0000_H
- TAUDnCDRm(slave 3) = 000B_H

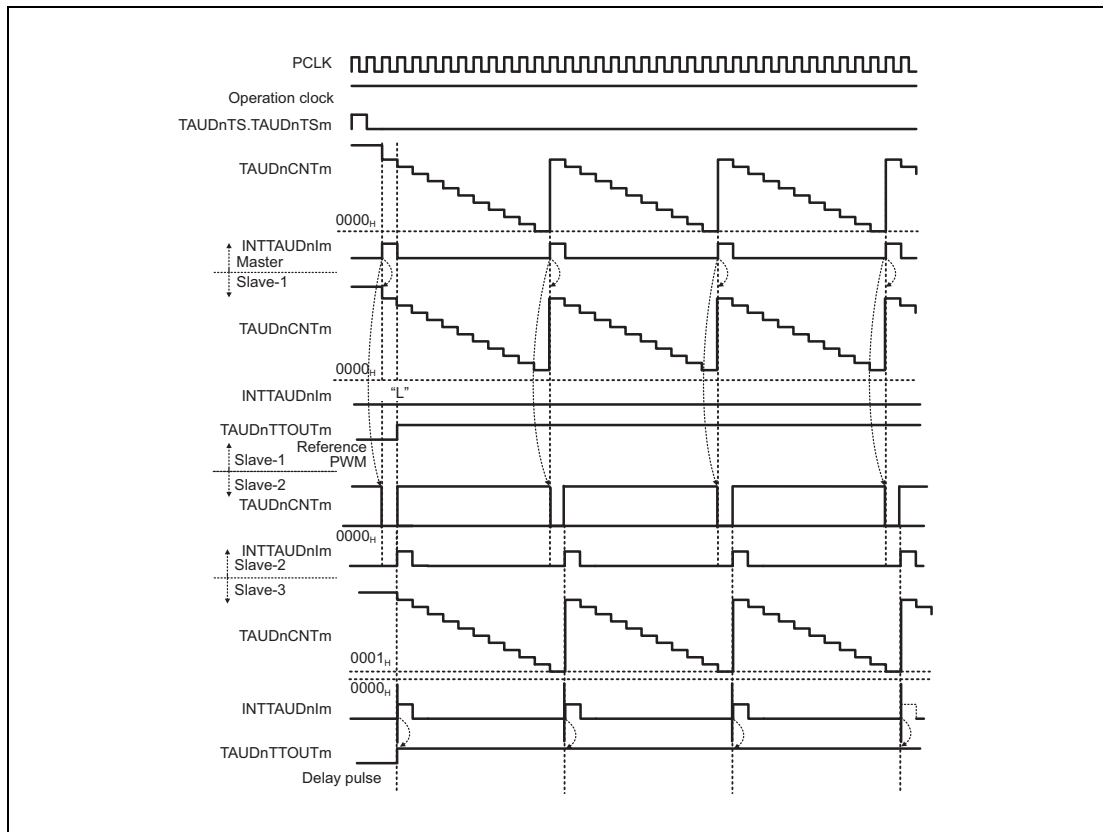


Figure 23.97 Duty Cycle (slave 3) = 100%

- If the value of TAUDnCDRm (slave 1 and 3) is higher than the value of TAUDnCDRm (master), the counter of slave channel 1 cannot reach 0000_H and cannot generate interrupts. TAUDnTTOUTm of channels 1 and 3 remain in the active state.

(2) TAUDnTTOUTm (slave 1)= TAUDnTTOUTm (slave 3)

The following values apply to **Figure 23.98**.

- TAUDnCDRm (master) = 000A_H
- TAUDnCDRm (slave 1) = 0005_H
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = 0005_H

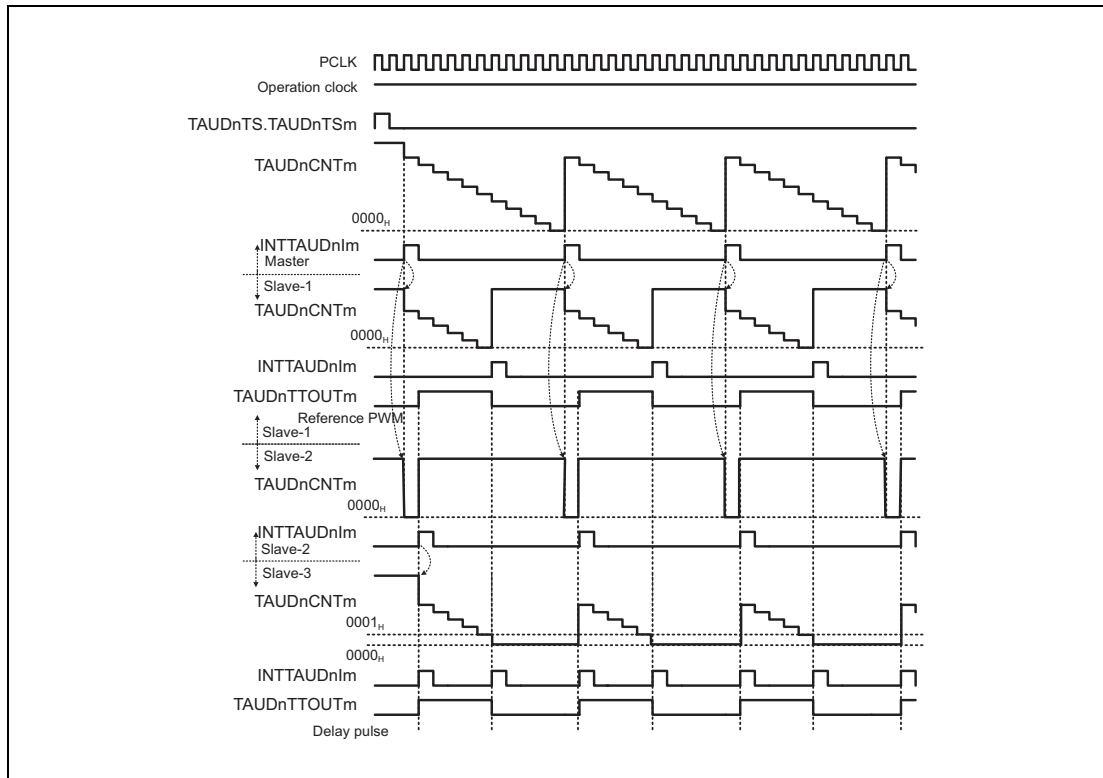


Figure 23.98 TAUDnTTOUTm (Slave 1)= TAUDnTTOUTm (Slave 3)

- If TAUDnCDRm (slave 2) = 0000_H, the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

23.13.4 Offset Trigger Output Function

23.13.4.1 Overview

Summary

This function generates triangle PWM outputs by using a master channel and slave channel. This sets the pulse cycle (frequency) of TAUDnTTOUTm.

The pulse cycle is set by detecting an effective input edge of the master channel. The pulse width is specified by the slave channel.

Prerequisites

- Two channels
- The operating mode for master channels should be set to capture mode. (See **Table 23.161, Contents of the TAUDnCMORm register for the Master Channel of the Triangle PWM Output Function.**)
- The operating mode for slave channels should be set one-count mode. (See **Table 23.165, Contents of the TAUDnCMORm register for the slave channel of the triangle PWM output function.**)
- The output mode for slave channels should be set to synchronous channel output mode 1. (See **Section 23.7, Channel Output Modes.**)
- TAUDTTOUTm is not used for master channels when this function is used.

Functional description

The counters are started by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This in turn sets TAUDnTE.TAUDnTEM, enabling count operation. The counter of the master channel (TAUDnCNTm) starts to count up from 0000_H.

- Master channel:
When an effective TAUDTTINm input edge is detected, the current value of the counter (TAUDnCNTm) is loaded to the data register (TAUDnCDRm) of the master channel. INTTAUDnIm is generated and the counter restarts to count up from 0000_H.
- Slave channels:
When INTTAUDnIm is generated on the master channel, the TAUDTTOUTm (slave) signal is set and the counter of the slave channel is triggered. The current value of the TAUDnCDRm (slave) is loaded to TAUDnCNTm (slave) and the counter starts to count down from the TAUDnCDRm value.
When the counter reaches 0000_H (duty time is elapsed), INTTAUDnIm is generated and the INTTAUDnIm signal is reset. The counter is returned to FFFF_H and waits for the next INTTAUDnIm of the master channel.

Setting TAUDnTT.TAUDnTTm of the master/slave channel to 1 stops the counter operation. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of the master/slave channel stop, but retain their respective values. Setting TAUDnTS.TAUDnTSM to 1 restarts the count operation.

23.13.4.2 Equations

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

Duty cycle [%] = [TAUDnCDRm (slave) / (TAUDnCDRm (master) + 1)] × 100

- Duty cycle = 0%
TAUDnCDRm (slave) = 0000_H
- Duty cycle = 100%
TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

23.13.4.3 Block Diagram and General Timing Diagram

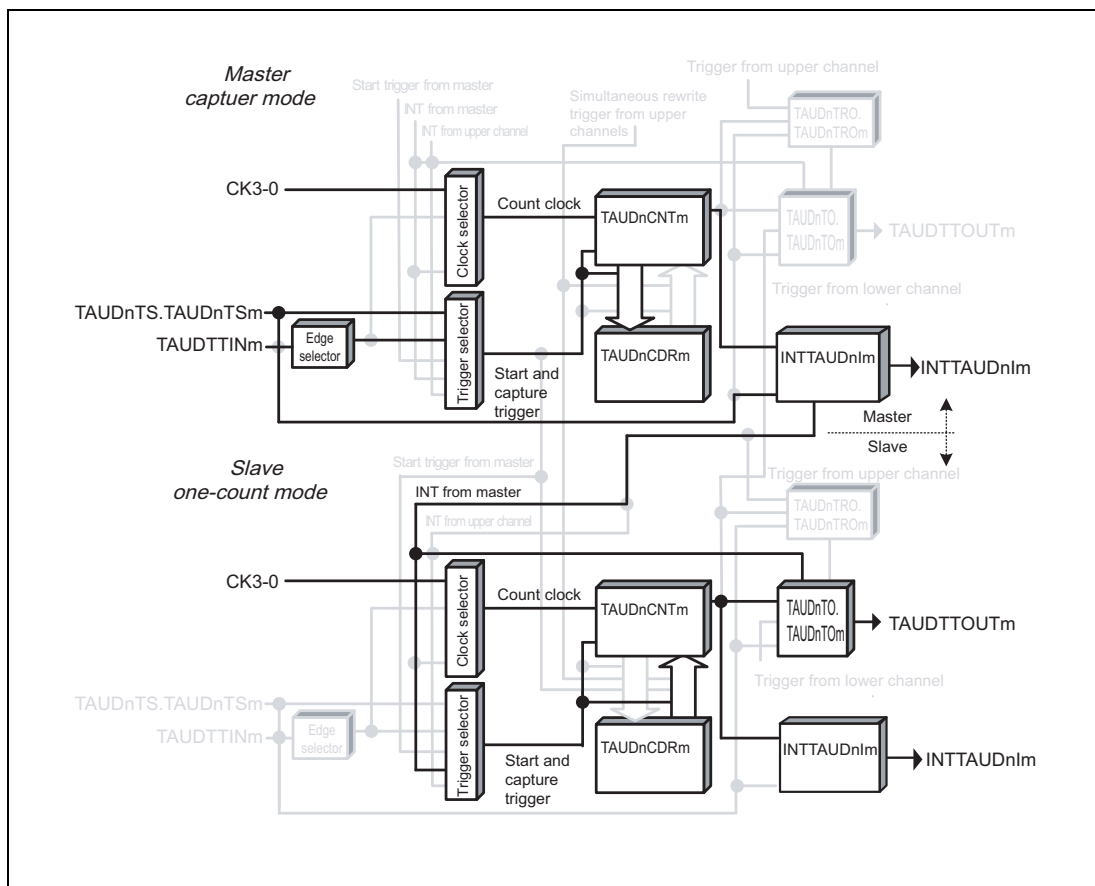


Figure 23.99 Block Diagram of Offset Trigger Output Function

The following settings apply to the general timing diagram.

- When a falling edge is detected (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

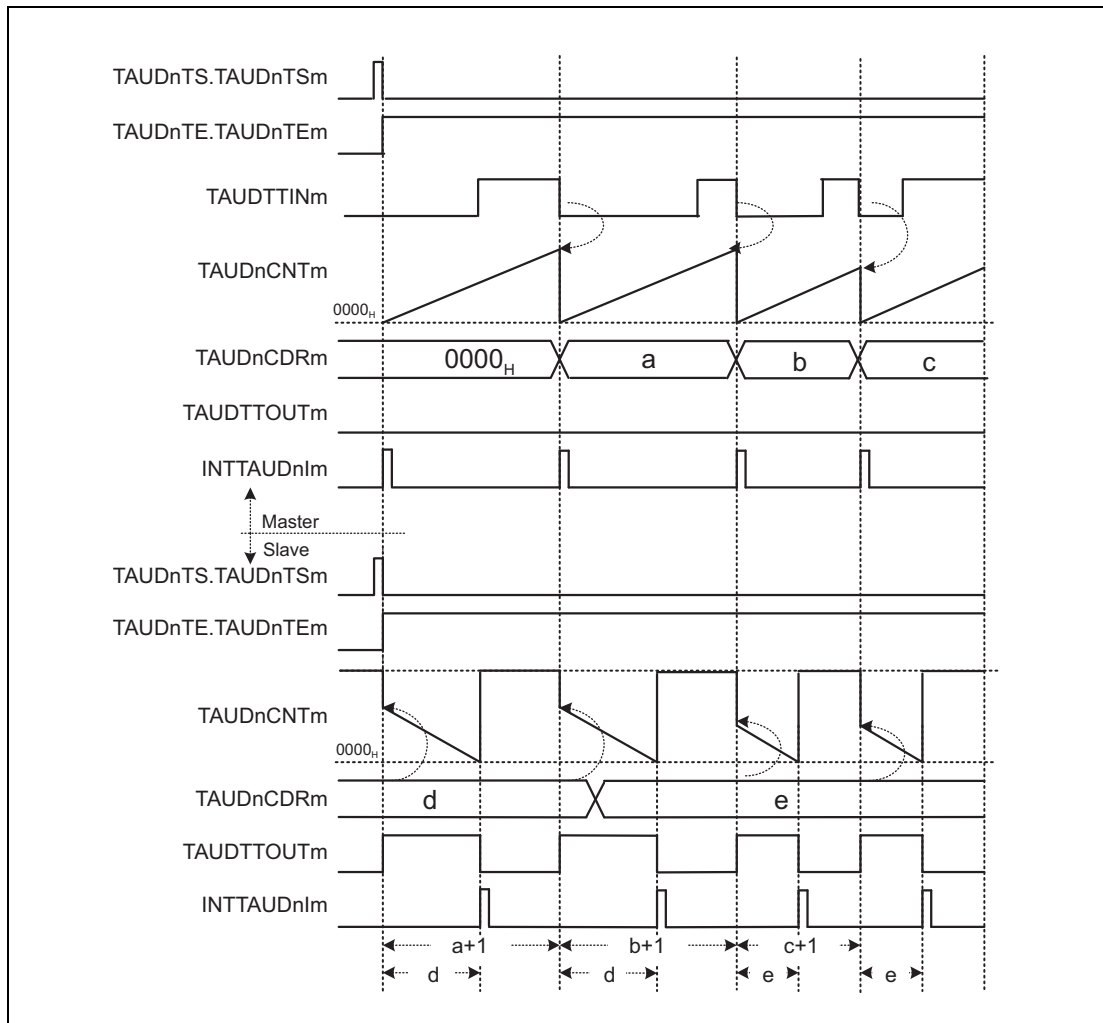


Figure 23.100 General Timing Diagram of Offset Trigger Output Function

NOTE

The TAUDTTOUTm signal from slave channel rises 1 cycle of the counter clock after the rising edge of INTTAUDnIm from the master channel.

23.13.4.4 Register Settings for Master Channels

(1) TAUDnCMORm for master channels

| | | | | | | | | | | | | | | | | |
|-------------------|----------------|-----|----------------|-----|----------|---------------|-----|-----|----------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS [1:0] | | TAUDnCCS [1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS [1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.153 Contents of the TAUDnCMORm register for the Master Channel of the Offset Trigger Output Function

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 15, 14 | TAUDnCKS [1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 001: Uses an effective TAUDTTINm input edge signal as a start trigger |
| 7, 6 | TAUDnCOS[1:0] | 11: Updates on detection of an effective TAUDTTINm input edge and occurrence of a counter overflow: <ul style="list-style-type: none"> – Detection of an effective TAUDTTINm input edge: A counter value is written to TAUDnCDRm. – Occurrence of an overflow; FFFF is written to TAUDnCDRm. The next effective TAUDTTINm input edge to be detected TAUDnCSRm.TAUDnOVF is set when a counter overflow occurs and cleared by the CPU instruction (TAUDnCSCm.TAUDnCLOV is set to 1). |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0010: Capture mode |
| 0 | TAUDnMD0 | 1: INTTAUDnIm is generated at the beginning of operation. |

(2) TAUDnCMURm for master channels

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 23.154 Contents of the TAUDnCMURm register for the Master Channel of the Offset Trigger Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: A falling edge is detected. 01: A rising edge is detected. 10: Falling and rising edges are detected. 11: Setting prohibited |

(3) Channel output mode for master channels

TAUDnTOE.TAUDnTOEm is set to 0 since channel output mode is not used in this function.

(4) Simultaneous rewrite for master channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers must be set to 0.

Table 23.155 Simultaneous Rewrite Settings for Master Channels of Offset Trigger Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Disables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | When simultaneous rewrite is disabled, set this bit to 0 (TAUDnRDE.TAUDnRDEm = 0). |
| TAUDnRDM.TAUDnRDMm | |
| TAUDnRDC.TAUDnRDCm | |

23.13.4.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------|-----|-------------------|-----|--------------|---------------|-----|-----|-------------------|-----|---|--------------|-----|-----|-----|--------------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS [1:0] | | TAUDnCCS [1:0] | | TAUDn MAS | TAUDnSTS[2:0] | | | TAUDnCOS [1:0] | | — | TAUDnMD[4:1] | | | | TAUDn MD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.156 Contents of TAUDnCMORm Register for Slave Channel of Offset Trigger Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 100: Up/down output trigger signal of master channel |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1001: One-count mode |
| 0 | TAUDnMD0 | 1: Enables start trigger detection while counting. |

(2) TAUDnCMURm for slave channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.157 Contents of the TAUDnCMURm Register for Slave Channel of Offset Trigger Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channels**Table 23.158 Control Bit Settings in Synchronous Channel Output Mode 1**

| Bit Name | Setting |
|--|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel operation |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 1 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm TAUDnTDL.TAUDnTDLm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0 |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for slave channels

Simultaneous write registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers must be set to 0.

Table 23.159 Simultaneous Rewrite Settings for Slave Channels of Offset Trigger Output Function

| Bit Name | Setting |
|--|--|
| TAUDnRDE.TAUDnRDEm | 0: Disables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm TAUDnRDM.TAUDnRDMm TAUDnRDC.TAUDnRDCm | When simultaneous rewrite is disabled, set this bit to 0 (TAUDnRDE.TAUDnRDEm = 0). |

23.13.4.6 Operating Procedure for Offset Trigger Output Function

Table 23.160 Operating Procedure for Offset Trigger Output Function

| | Operation | TAUDn Status |
|-------------------------|---|---|
| Initial Channel Setting | Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.4.4, Register Settings for Master Channels. | Channel operation is stopped. |
| | Slave channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.4.5, Register Settings for Slave Channels. | |
| | The TAUDnCDRm register of master channel operates as a capture register. Set the value of the TAUDnCDRm register of slave channel. | |
| Start Operation | Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. <ul style="list-style-type: none"> TAUDnCNTm (master) starts to count up. The TAUDnCDRm value is loaded to TAUDnCNTm (slave) to perform counting down. INTTAUDnIm is generated on the master channel and TAUDnTTOUTm (slave) is set. |
| | TAUDnCDRm can be changed at any time. TAUDnCSCm.TAUDnCLOV can be set to 1. TAUDnCDRm of slave channel can be changed after INTTAUDnIm is generated. TAUDnCNT.TAUDnCNTm and TAUDnCSRm are readable at any time. | When the counter of master channel reaches 0000 _H : <ul style="list-style-type: none"> INTTAUDnIm (slave) occurs. TAUDnTTOUTm (slave) is reset and count operation stops on the slave channel. When a TAUDTTINm input edge is detected on the master channel: <ul style="list-style-type: none"> INTTAUDnIm (master) is generated. TAUDnCNTm (master) is reset to 0000_H and continues to count. TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down. TAUDTTOUTm (slave) is set. |
| During Operation | | |
| Stop Operation | Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.13.4.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the specific timing diagram.

- When a falling edge is detected (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

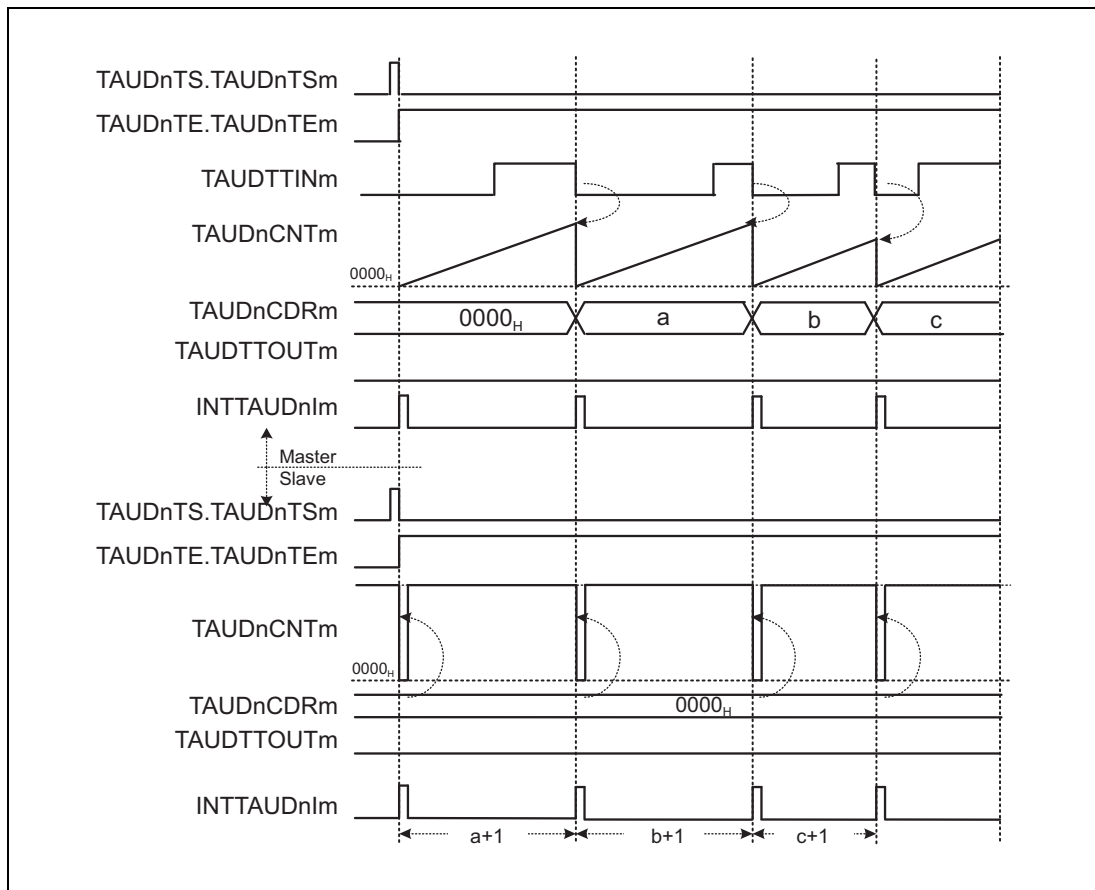


Figure 23.101 TAUDnCDRm (Slave) = 0000_H

- When TAUDnCDRm (slave) is 0000_H, 0000_H is written to TAUDnCNTm every time master channel generates an interrupt (INTTAUDnIm). Thus TAUDnCNTm cannot start to count operation. TAUDTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the TAUDnCDRm value is reloaded. Slave and master channels generate interrupts in the same cycle.

(2) Duty cycle = 100%

The following settings apply to the specific timing diagram.

- When a falling edge is detected (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

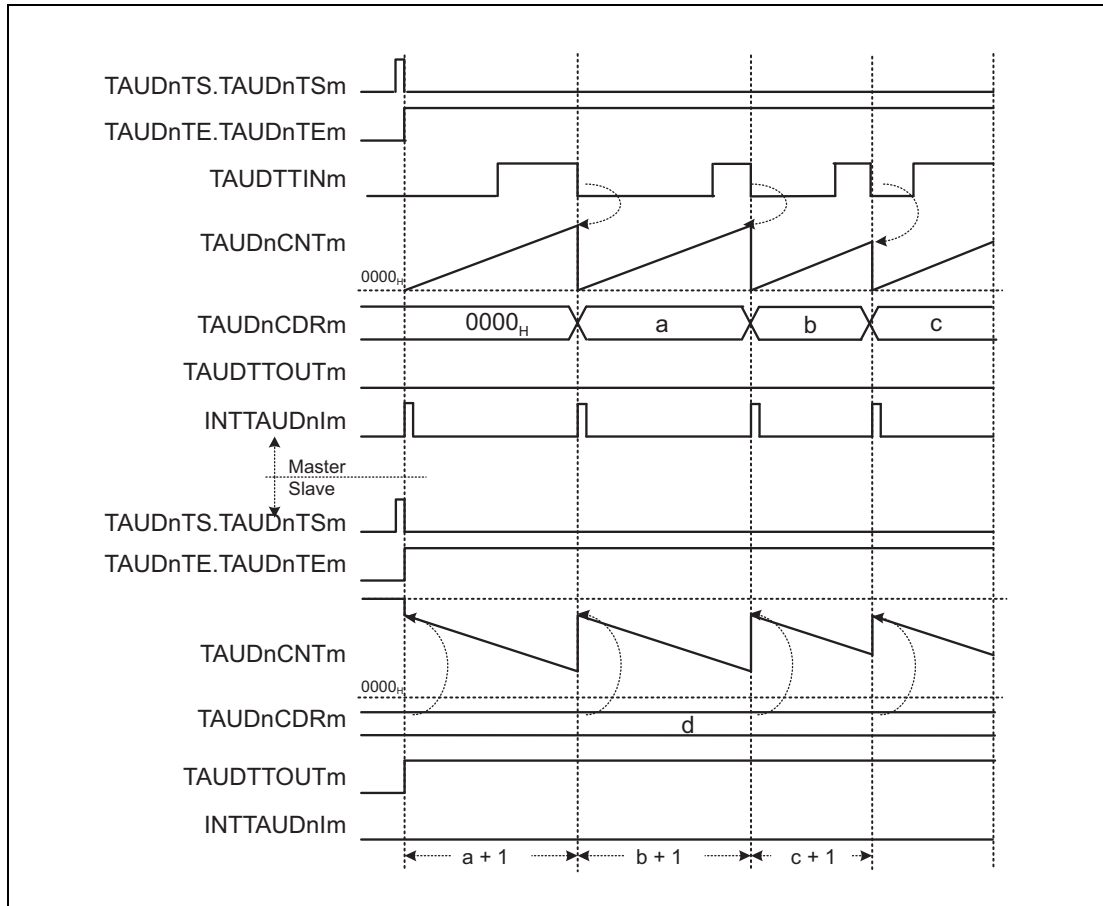


Figure 23.102 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- When TAUDnCDRm (slave) value exceeds the interval of an effective input edge, the counter of slave channel does not reach 0000H, and an interrupt is not generated. TAUDTTOUTm remains active.

23.13.5 A/D Conversion Trigger Output Function Type 1

23.13.5.1 Overview

Summary

This function is identical to **Section 23.13.1, PWM Output Function**, except that TAUDnTTOUTm is not output.

This function is enabled by setting the channel output mode for the slave to independent channel output mode controlled by software.

23.13.5.2 Block Diagram and General Timing Diagram

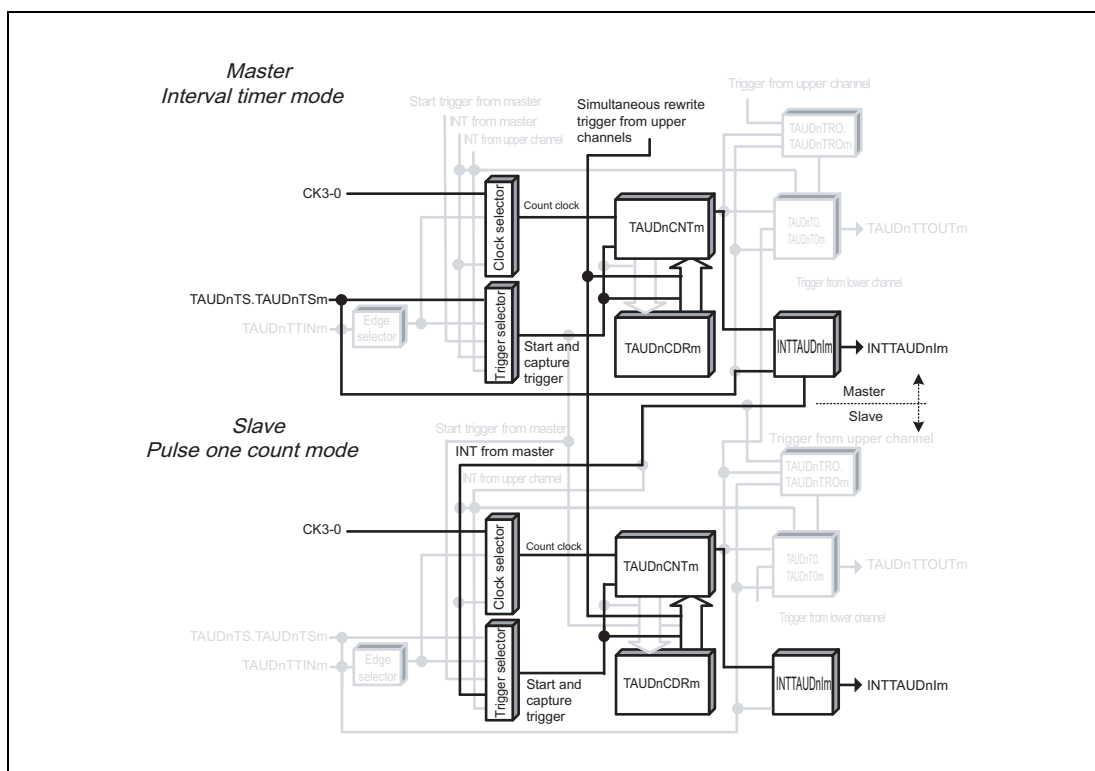


Figure 23.103 Block Diagram of A/D Conversion Trigger Output Function Type 1

23.13.5.3 General Timing Diagram

The following settings apply to the general timing diagram.

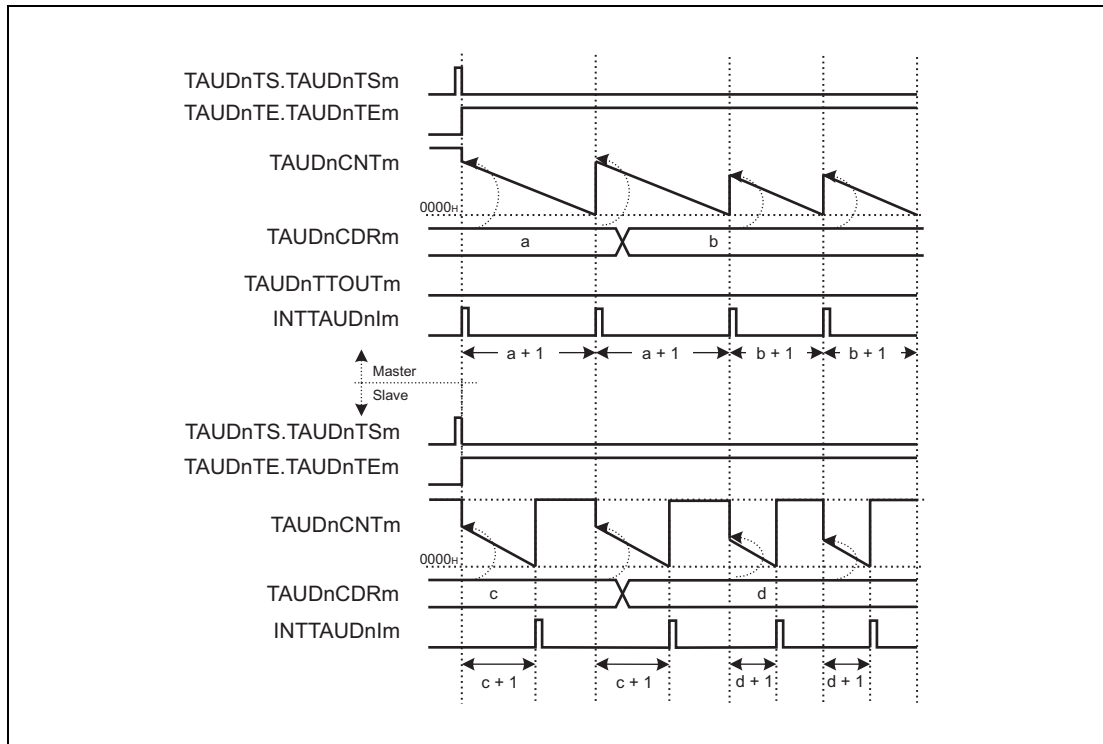


Figure 23.104 General Timing Diagram of A/D Conversion Trigger Output Function Type 1

23.13.6 Triangle PWM Output Function

23.13.6.1 Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUDnTTOUTm to be set using the master and slave channels respectively.

The master channel generates a carrier cycle. The first cycle of master channel controls the down status and the second cycle controls the up status of the slave counter.

Prerequisites

- Two channels
- The operating mode for master channels should be set to interval timer mode. (See **Table 23.161, Contents of the TAUDnCMORm register for the Master Channel of the Triangle PWM Output Function.**)
- The operating mode for slave channels should be set up/down count mode. (See **Table 23.165, Contents of the TAUDnCMORm register for the slave channel of the triangle PWM output function.**)
- The channel output mode for master channels should be set to independent channel output mode 1. (See **Section 23.7, Channel Output Modes.**)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See **Section 23.7, Channel Output Modes.**)
- The following settings allow TAUDnTTOUTm signal to be at high level during the down status of a carrier cycle.
 - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
 - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

Functional description

The counters are started by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for every channel. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation.

The current values of TAUDnCDRm (master and slave) are loaded into TAUDnCNTm (master and slave) and the counters start counting down from these values. When the TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and TAUDnTTOUTm signal of master toggles.

- Master channel:

When the counter of master channel reaches 0000_H (pulse cycle time has elapsed), INTTAUDnIm is generated and the TAUDnTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and counts down.

- Slave channels:

The INTTAUDnIm of master channel triggers the counter of the slave channel:

- If the slave counter is counting down, the count direction changes.
- If the slave counter is counting up, TAUDnCDRm value is reloaded and the counter starts to count down.

When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUDnIm is generated and the TAUDnTTOUTm (slave) signal is set/reset.

The counter continues count-up/-down and waits for the next INTTAUDnIm of master channel.

Setting TAUDnTOL.TAUDnTOLm allows TAUDnTTOUTm signal switching between normal phase and reverse phase during operation.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values.

Conditions

This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.

23.13.6.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

0000_H ≤ TAUDnCDRm (master) < FFFF_H

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle [%] =

$(\text{TAUDnCDRm (master)} + 1 - \text{TAUDnCDRm (slave)}) / (\text{TAUDnCDRm (master)} + 1) \times 100$

- Duty cycle = 100%

TAUDnCDRm (slave) = 0000_H

- Duty cycle = 0%

TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

23.13.6.3 Block Diagram and General Timing Diagram

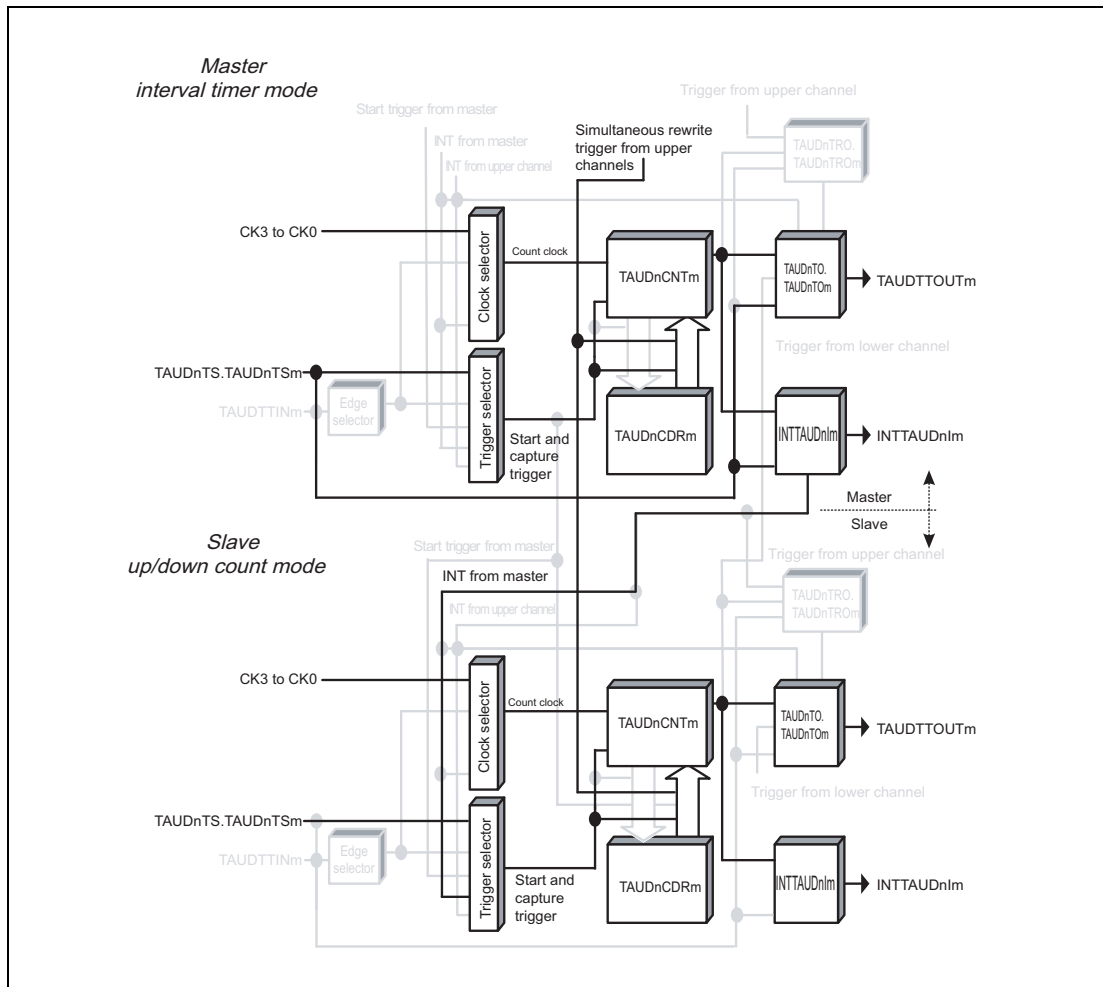


Figure 23.105 Block Diagram of Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
 - INTTAUDnIm generated at the beginning of operation.
(TAUDnCMORm.TAUDnMD0 = 1)

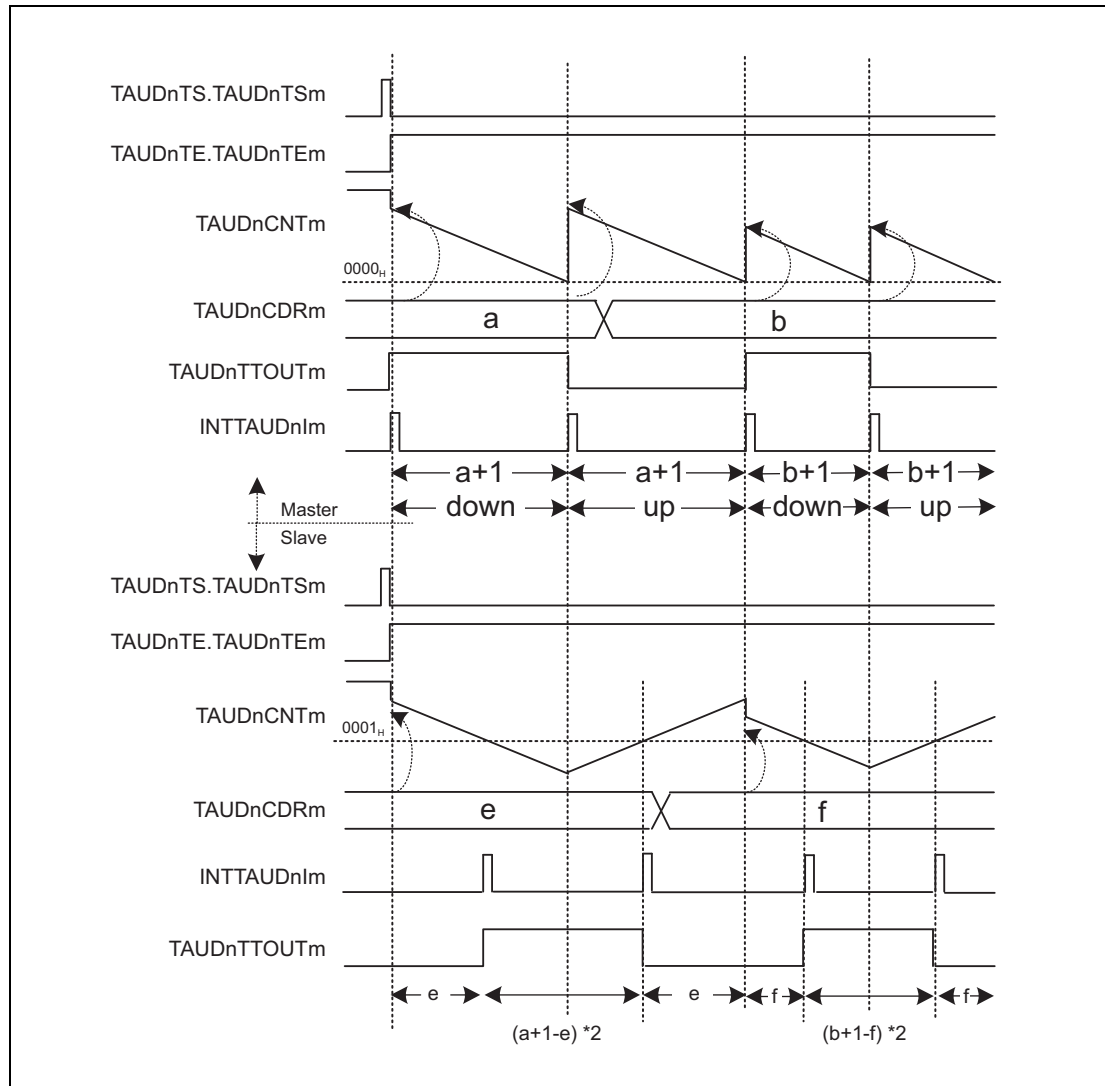


Figure 23.106 General Timing Diagram of Triangle PWM Output Function

23.13.6.4 Register Settings for Master Channels

(1) TAUDnCMORm for master channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.161 Contents of the TAUDnCMORm register for the Master Channel of the Triangle PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated to toggle TAUDnTTOUtm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDnTTOUtm is toggled at the beginning of operation. |

(2) TAUDnCMURm for master channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.162 Contents of the TAUDnCMURm register for the Master Channel of the Triangle PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for master channels**Table 23.163 Control Bit Settings in Independent Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|--|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: The setting is disabled in toggle mode. (The value is the initial value after reset.) |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROM | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for master channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.164 Simultaneous Rewrite Settings for Master Channels of Triangle PWM Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not use the channel to generate the simultaneous rewrite trigger. |

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

23.13.6.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.165 Contents of the TAUDnCMORm register for the slave channel of the triangle PWM output function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 111: Up/down output trigger signal of master channel |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1001: Up/down count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. |

(2) TAUDnCMURm for slave channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.166 Contents of the TAUDnCMURm register for the slave channel of the triangle PWM output function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channels**Table 23.167 Control Bit Settings in Synchronous Channel Output Mode 2**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel operation |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for slave channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.168 Simultaneous Rewrite Settings for Slave Channels of Triangle PWM Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not use the channel to generate the simultaneous rewrite trigger. |

23.13.6.6 Operating Procedure for Triangle PWM Output Function

Table 23.169 Operating Procedure for Triangle PWM Output Function

| | Operation | TAUDn Status |
|-------------------------|--|---|
| Initial Channel Setting | Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.6.4, Register Settings for Master Channels. | Channel operation is stopped. |
| | Slave channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.6.5, Register Settings for Slave Channels. | |
| | Set the value of TAUDnCDRm register of every channel. | |
| Restart | Start Operation | <p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p> <p>TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.</p> |
| During Operation | <p>TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnTOL.TAUDnTOLm can be changed.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p> | <p>TAUDnCDRm value of master and slave channels is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000_H:</p> <ul style="list-style-type: none"> • INTTAUDnIm (master) occurs. • TAUDnTTOUTm (master) is toggled. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) or counting is started in opposite direction. <p>When TAUDnCNTm of slave channel reaches 0001_H:</p> <ul style="list-style-type: none"> • INTTAUDnIm (slave) occurs. • TAUDnTTOUTm (slave) is set in the count-down status or reset in count-up status. |
| | Stop Operation | <p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p> <p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.</p> |

23.13.6.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
 - INTTAUDnIm generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
 - TAUDnCDRm = a = 5_H
- Slave channels:
 - TAUDnCDRm = 6_H

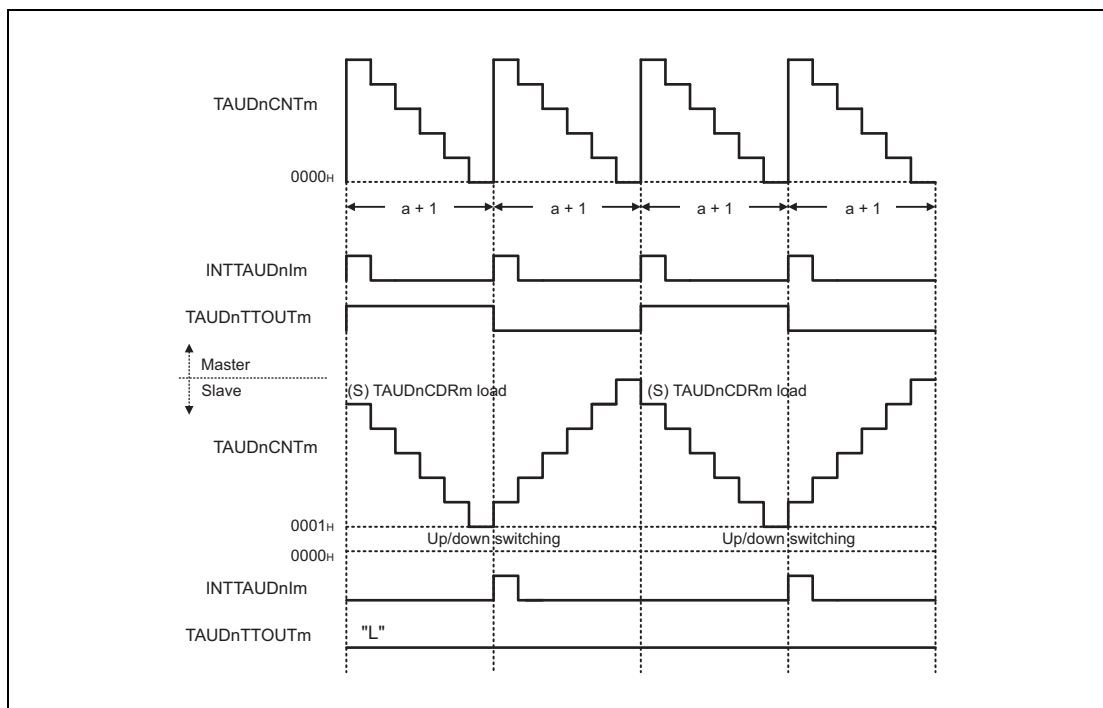


Figure 23.107 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave) value is greater than TAUDnCDRm (master) value + 1, INTTAUDnIm of the slave channel is not generated while counting down. TAUDnTTOUTm remains low because there is no set signal to be detected.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
 - INTTAUDnIm generated at the beginning of operation. ($\text{TAUDnCMORm.TAUDnMD0} = 1$)
 - $\text{TAUDnCDRm} = a = 5_{\text{H}}$
- Slave channels:
 - $\text{TAUDnCDRm} = 0_{\text{H}}$

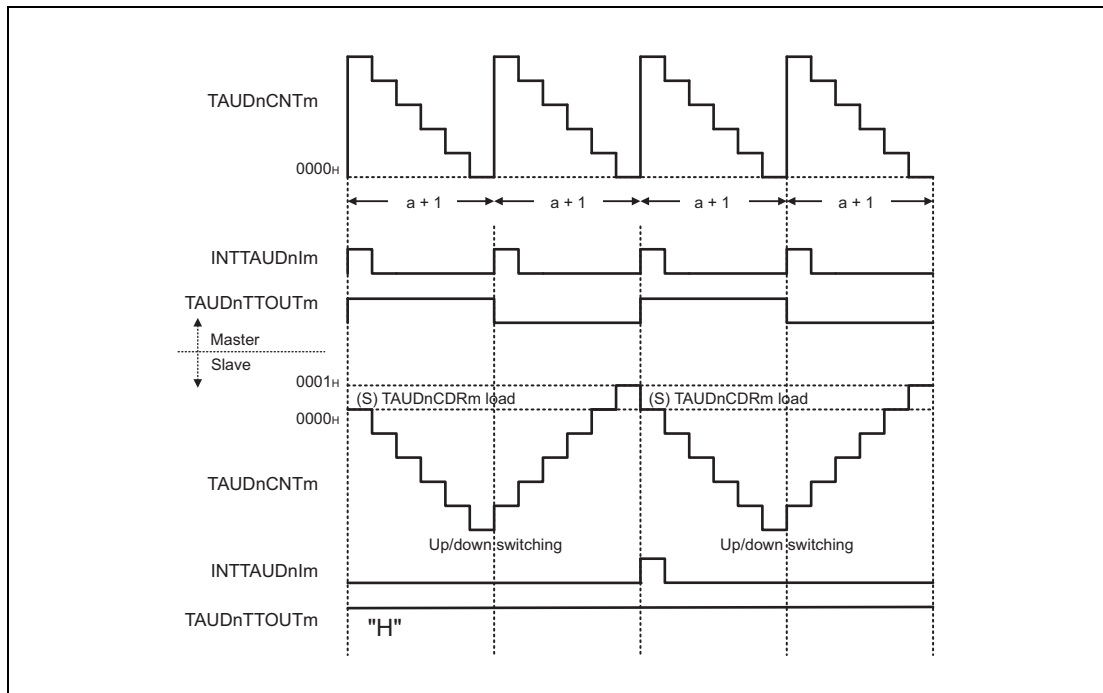


Figure 23.108 TAUDnCDRm (Slave) = 0000_H

- If $\text{TAUDnCDRm (slave)} = 0000_{\text{H}}$, INTTAUDnIm of slave channel is not generated while counting up. TAUDnTTOUTm remains high because there is no reset signal to be detected.

23.13.7 Triangle PWM Output Function with Dead Time

23.13.7.1 Overview

Summary

This function generates multiple triangle PWM outputs with a predefined dead time added by using a master and two or more slave channels. The resulting PWM signals with dead time are output via TAUDnTTOUTm of the slave channels 2 and 3, enabling the pulse cycle (frequency) and the duty cycle of TAUDnTTOUTm to be set using the master and slave channels.

Carrier cycles are generated on master channel. The first pulse controls the down status of slave counter and the second one controls the up status.

An interrupt on slave 2 causes TAUDnTTOUTm of slave channels to be set/reset. Depending on the settings of TAUDnTDL.TAUDnTDLm, delay time is added to positive or negative logic side of the signal (i.e., whether TAUDnTTOUTm is set/reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. For slave channels 2 and 3, select even-numbered channel CH (a) and odd-numbered channel CH (a + 1).
- The operating mode for master channels should be set to interval timer mode. (See **Table 23.171, Contents of the TAUDnCMORm register for the Master Channel of the Triangle PWM Output Function with dead time**)
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even-numbered channel (a), and slave channel 3 is an odd-numbered channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operating mode for slave channel 2 should be set to up/down count mode (See **Table 23.175, Contents of the TAUDnCMORm register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time**). Slave channel 2 should be an even-numbered channel.
- The operating mode for slave channel 3 should be set to one-count mode (See **Table 23.179, Contents of the TAUDnCMORm register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time**). Slave channel 3 should be an odd-numbered channel.
- The channel output mode for master channels should be set to independent channel output mode 1. (See **Section 23.7, Channel Output Modes**)
- The output mode for slave channels 2 and 3 should be set to synchronous channel output mode 2 with dead time output. (See **Section 23.7, Channel Output Modes**)

- The following settings make a TAUDnTTOUTm signal at high level during the down status of the carrier cycle:
 - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0. (recommended setting)
 - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

NOTE

The triangle PWM output function with dead time does not use slave channel 1. Slave channel 1 can be used as a separate timer (independent function).

Functional description

The counter starts by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and the master's TAUDnTTOUTm signal is toggled.

- Master channel:
When the counter of master channel reaches 0000_H, an INTTAUDnIm is generated and the TAUDnTTOUTm signal is toggled. The TAUDnCDRm value is reloaded to continue countdown.
- Slave channel 2:
If INTTAUDnIm is generated on the master channel, the counter of slave channel 2 is triggered.
 - If the slave counter is counting down, the counting direction changes.
 - If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts to count down.

The counter continues to count down/up and waits for the next INTTAUDnIm of master channel. When the counter value of slave channel 2 reaches 0001_H, INTTAUDnIm is generated.

- Slave channel 3:
If INTTAUDnIm is generated on slave channel 2, the counter of slave channel 3 is triggered. The current value of TAUDnCDRm (slave 3) is loaded into TAUDnCNTm (slave 3) and the counter starts to count down from the TAUDnCDRm value.
When the counter reaches 0000_H, INTTAUDnIm occurs. The counter returns to FFFF_H and waits for the next INTTAUDnIm of slave channel 2.

As described in **Table 23.170, Operation of TAUDnTTOUTm upon Occurrence of an Interrupt on Slave Channel 2**, the set/reset timing (right after occurrence of an interrupt or after dead time has elapsed) depends on the TAUDnTDL.TAUDnTDLm setting of the corresponding channel.

The setting of TAUDnTOL.TAUDnTOLm also determines whether a high level signal (TAUDnTOL.TAUDnTOLm = 0) or a low level signal (TAUDnTOL.TAUDnTOLm = 1) is output from the corresponding channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values.

TAUDnTTOUTm can be 100% output by setting the TAUDnCDRm value of slave channel 2 to 0000_H.

Conditions

This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.

TAUDnTOL.TAUDnTOLm and TAUDnTDL.TAUDnTDLm should be set before start of count operation. Slave channels 2 and 3 should have the opposite settings of TAUDnTOL.TAUDnTOLm or TAUDnTDL.TAUDnTDLm.

Table 23.170 Operation of TAUDnTTOUTm upon Occurrence of an Interrupt on Slave Channel 2

| TAUDnTDL. TAUDnTDLm | Count Direction of Slave Channel 2 upon Occurrence of Interrupt | TAUDnTTOUTm Set/Reset Timing |
|------------------------|--|------------------------------------|
| 0 | Down | Set after elapse of dead time |
| | Up | Reset right after interrupt occurs |
| 1 | Down | Set right after interrupt occurs |
| | Up | Reset after elapse of dead time |

23.13.7.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

0000_H ≤ TAUDnCDRm (master) < FFFF_H

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (normal phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (reverse phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 + (TAUDnCDRm (slave 3) + 1)] × count clock cycle

23.13.7.3 Block Diagram and General Timing Diagram

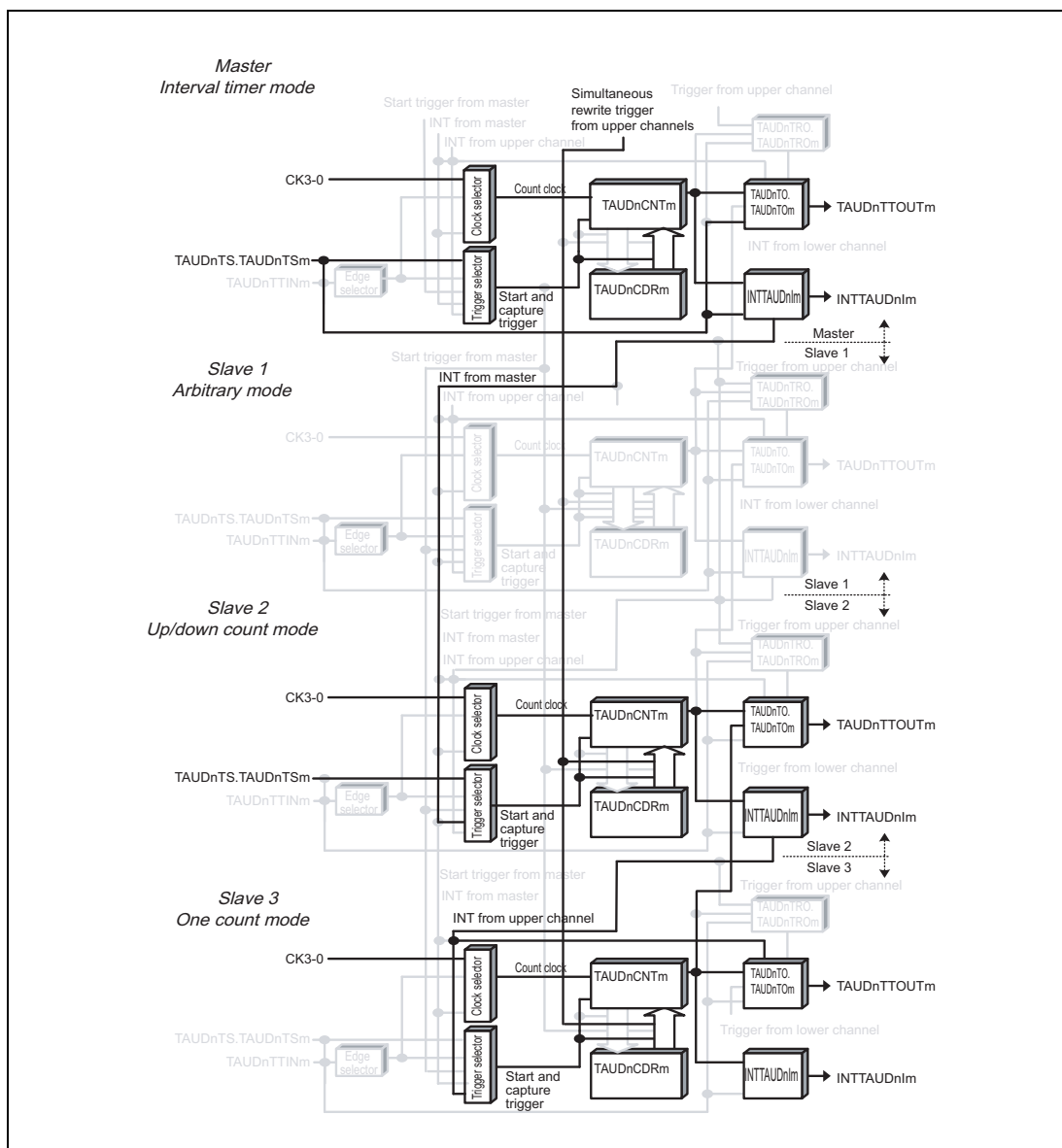


Figure 23.109 Block Diagram of Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
 - INTTAUDnIm generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Slave channel 2:
 - INTTAUDnIm not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
 - TAUDnTDL.TAUDnTDLm = 0
 - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

- Slave channel 3:
 - Start trigger detection enabled during counting ($TAUDnCMORm.TAUDnMD0 = 1$)
 - $TAUDnTDL.TAUDnTDLm = 1$
 - Positive logic ($TAUDnTOL.TAUDnTOLm = 0$)

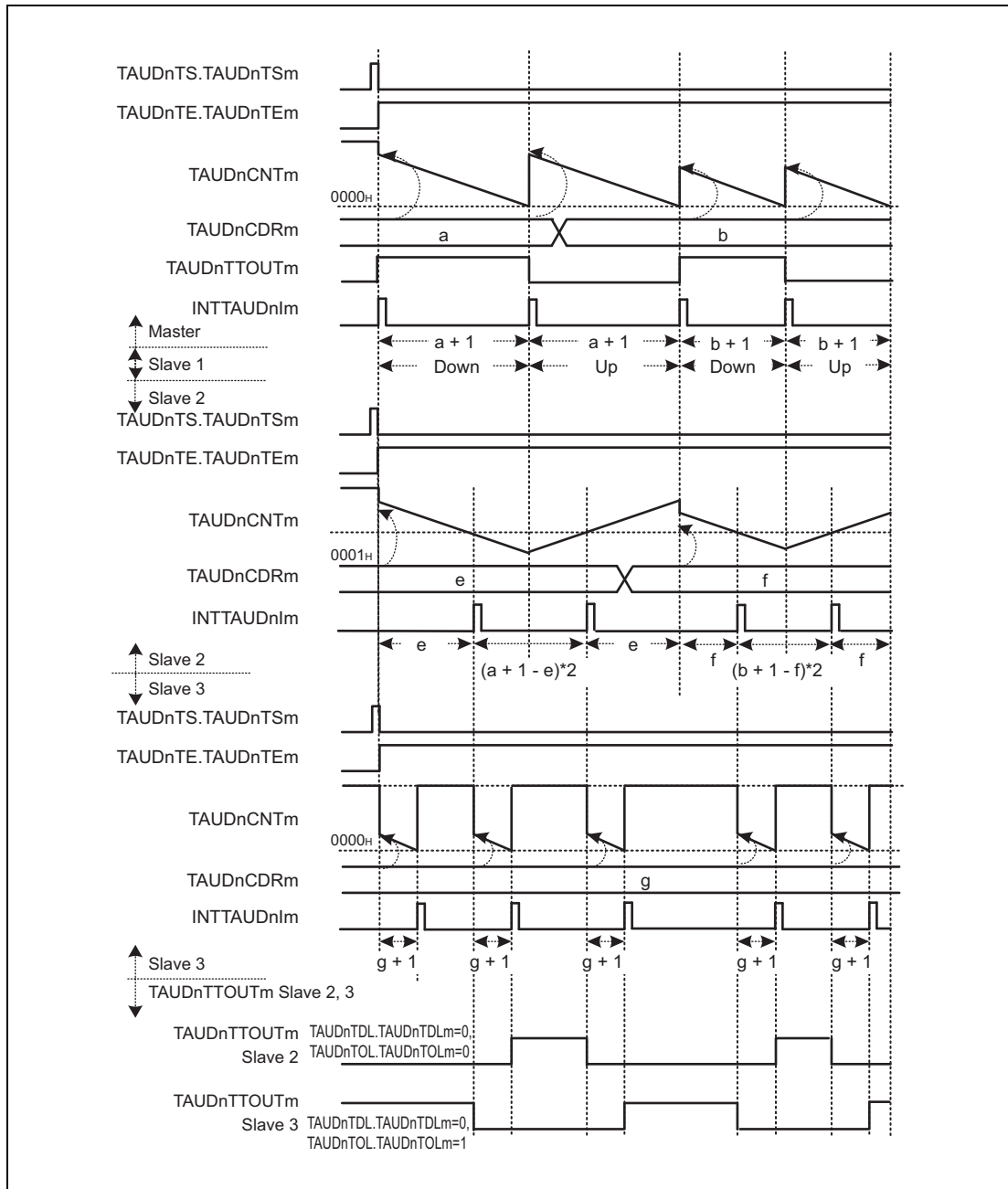


Figure 23.110 General Timing Diagram of Triangle PWM Output Function with Dead Time

23.13.7.4 Register Settings for Master Channels

(1) TAUDnCMORm for master channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.171 Contents of the TAUDnCMORm register for the Master Channel of the Triangle PWM Output Function with dead time

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated and TAUDnTTOUTm is not toggled at the beginning of operation. 1: INTTAUDnIm is generated and TAUDnTTOUTm is toggled at the beginning of operation. |

(2) TAUDnCMURm for master channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.172 Contents of the TAUDnCMURm register for the Master Channel of the Triangle PWM Output Function with dead time

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for master channels**Table 23.173 Control Bit Settings in Independent Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: The setting is disabled in toggle mode (The value after reset). |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROM | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for master channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.174 Simultaneous Rewrite Setting for Master Channels of Triangle PWM Output Function with Dead Time

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

23.13.7.5 Register Settings for Slave Channel 2

(1) TAUDnCMORm for slave channel 2

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.175 Contents of the TAUDnCMORm register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 111: Up/down output trigger signal of master channel |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1001: Up/down count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. |

(2) TAUDnCMURm for slave channel 2

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.176 Contents of the TAUDnCMURm register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channel 2**Table 23.177 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel operation |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 1: Enables dead time operation. |
| TAUDnTDM.TAUDnTDMm | 0: Adds dead time if an interrupt is detected on an even-numbered upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied. |
| TAUDnTDL.TAUDnTDLm | 0: Adds dead time to normal phase. 1: Adds dead time to reverse phase. |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROM | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from odd-numbered channels.

(4) Simultaneous rewrite for slave channel 2

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.178 Simultaneous Rewrite Settings for Slave Channel 2 of Triangle PWM Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.7.6 Register Settings for Slave Channel 3

(1) TAUDnCMORm for slave channel 3

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKs[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.179 Contents of the TAUDnCMORm register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKs[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKs[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 110: The up/down output trigger signal TAUDnUDCm of the master channel is the start trigger |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Enables start trigger detection while counting. |

(2) TAUDnCMURm for slave channel 3

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.180 Contents of the TAUDnCMURm register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channel 3**Table 23.181 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel operation |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 1: Enables dead time operation. |
| TAUDnTDM.TAUDnTDMm | 0: Adds dead time if an interrupt is detected on an even-numbered upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied. |
| TAUDnTDL.TAUDnTDLm | 0: Adds dead time to normal phase. 1: Adds dead time to reverse phase. |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROM | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | 0: Operation as a real-time output trigger channel is prohibited. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from even-numbered channels.

(4) Simultaneous rewrite for slave channel 3

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.182 Simultaneous Rewrite Settings for Slave Channel 3 of Triangle PWM Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.7.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 23.183 Operating Procedure for Triangle PWM Output Function with Dead Time

| | Operation | TAUDn Status |
|-------------------------|---|--|
| Initial Channel Setting | Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.7.4, Register Settings for Master Channels. | Channel operation is stopped. |
| | Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.7.5, Register Settings for Slave Channel 2. | |
| | Slave channel 3: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.7.6, Register Settings for Slave Channel 3. | |
| | Set the value of TAUDnCDRm register of every channel. | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSM of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1. |
| During Operation | TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. | TAUDnCDRm value of master channel and slave channel 2 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000 _H : |
| | TAUDnRDT.TAUDnRDTm can be changed during operation. | <ul style="list-style-type: none"> • INTTAUDnIm (master) occurs. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 2) or counting is started in opposite direction. When TAUDnCNTm of slave channel 2 reaches 0001 _H : <ul style="list-style-type: none"> • IINTTAUDnIm (slave 2) is generated. • TAUDnCDRm value of slave channel 3 is loaded into TAUDnCNTm perform counting down. When TAUDnCNTm of slave channel 3 reaches 0000 _H : <ul style="list-style-type: none"> • INTTAUDnIm is generated. |
| Stop Operation | Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.13.7.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram in **Figure 23.111**.

- Slave channel 2:
 - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
 - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

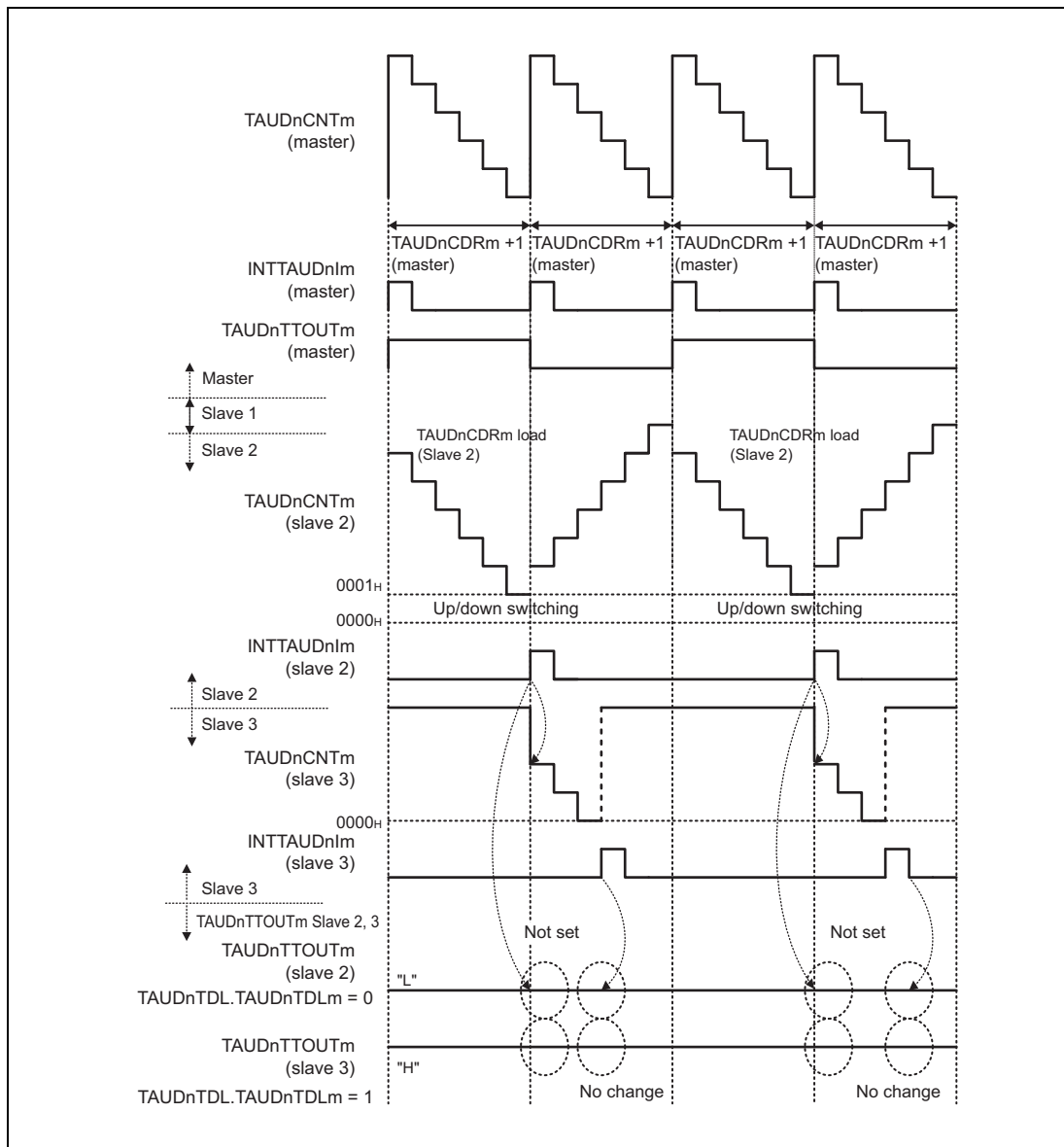


Figure 23.111 TAUDnCDRm (Slave 2) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave 2) is greater than TAUDnCDRm (master), the counter of slave channel does not reach 0000_H while counting down. Therefore, TAUDnTTOUtm signal is not set/reset and remains initial. This signal becomes a reset signal because an interrupt occurs on slave channel 2 during count-up operation.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram in **Figure 23.112**.

- Slave channel 2:
 - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
 - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

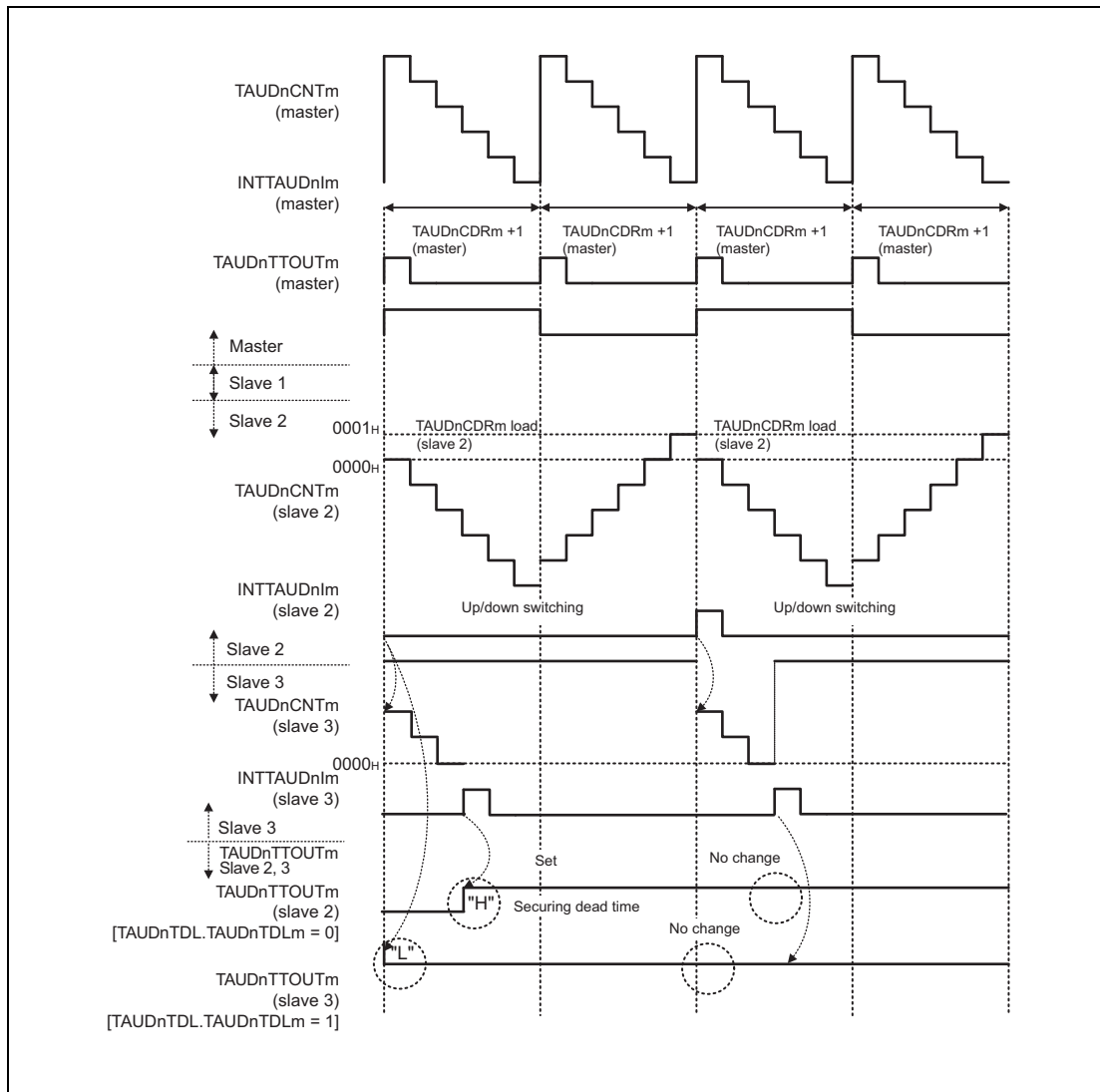


Figure 23.112 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave 2) = 0000_H, the slave channel counter does not reach 0001_H while counting up. Therefore, no INTTAUDnIm occurs during count-up operation.
 - The set conditions for a channel with TAUDnTDL.TAUDnTDLm = 0 are met after elapse of dead time. TAUDnTTOUTm is left in a newly set state even if a set/reset is made because no reset conditions are satisfied on such a channel.
 - Slave channel 3 in the above diagram is set when the counter starts. However, TAUDnTTOUTm is left in an initial state on the slave channel with TAUDnTDL.TAUDnTDLm = 1 because no reset conditions are satisfied on that channel.

23.13.8 A/D Conversion Trigger Output Function Type 2

23.13.8.1 Overview

Summary

This function is identical to **Section 23.13.6, Triangle PWM Output Function**, except that TAUDnTTOUTm is not output.

This function is enabled by setting channel output mode for the slave to independent channel output mode controlled by software.

23.13.8.2 Block Diagram and General Timing Diagram

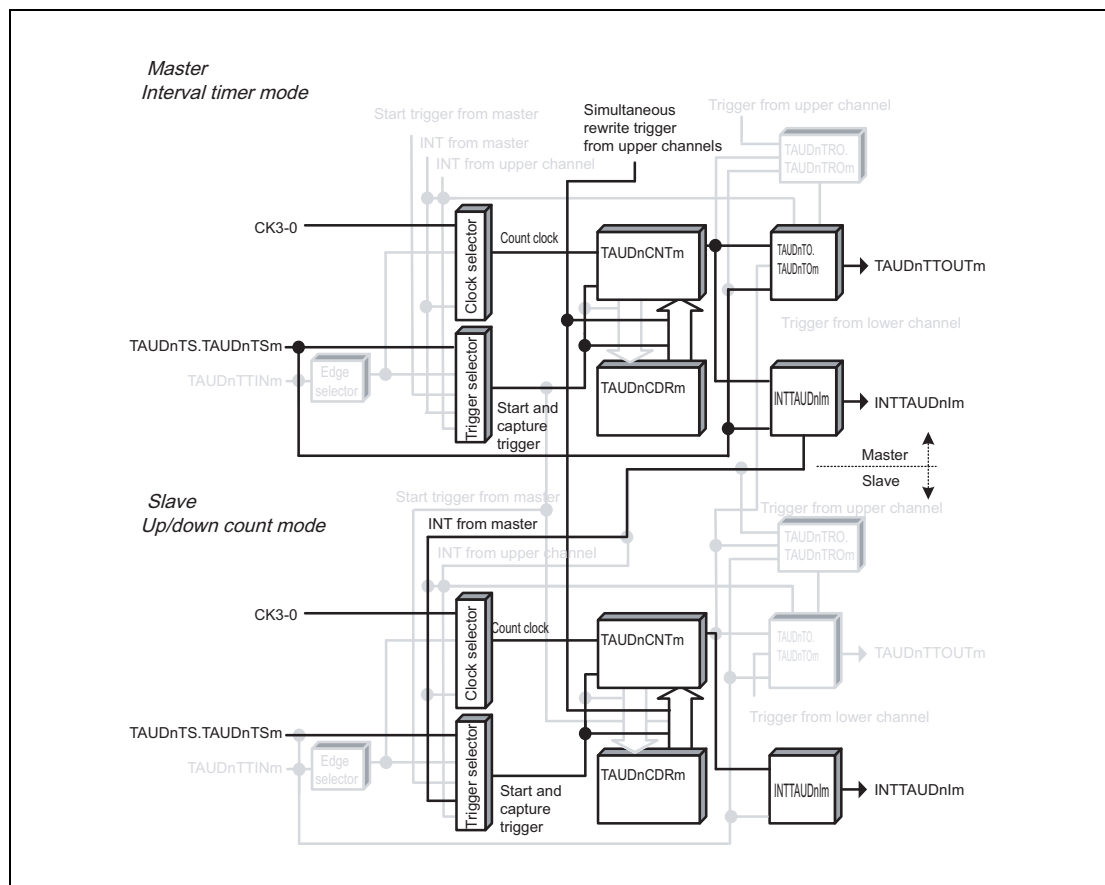


Figure 23.113 Block Diagram of A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
 - INTTAUDnIm generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

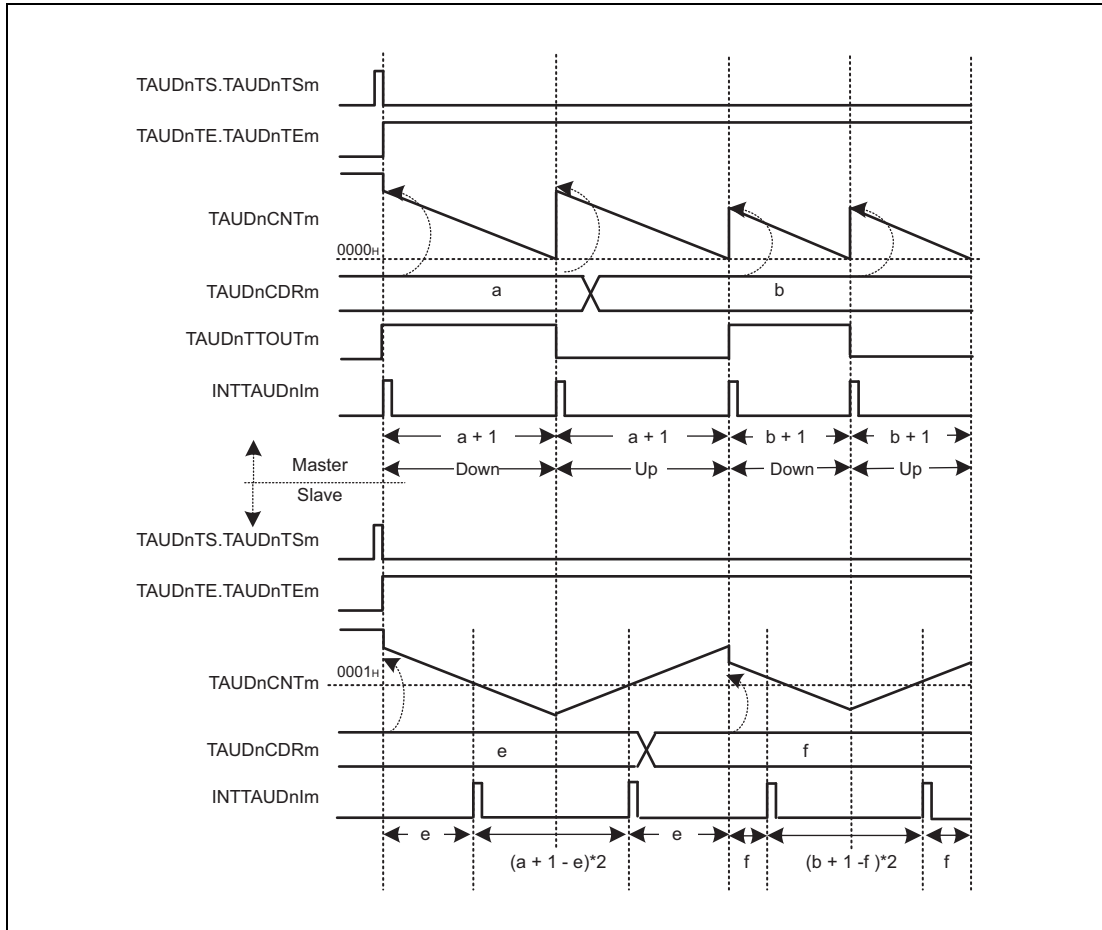


Figure 23.114 General Timing Diagram of A/D Conversion Trigger Output Function Type 2

23.13.9 Interrupt Request Signals Culling Function

23.13.9.1 Overview

Summary

This function divides the number of interrupts of the master channel by a specified value using a slave channel.

The interrupt request signals culling function is a sub function of the following functions:

- PWM Output Function (See **Section 23.13.1, PWM Output Function**)
- Triangle PWM Output Function (See **Section 23.13.6, Triangle PWM Output Function**)
- Triangle PWM Output Function with Dead Time
(See **Section 23.13.7, Triangle PWM Output Function with Dead Time**)

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode. (See **Table 23.184, Contents of the TAUDnCMORm register for the Master Channel of the Interrupt Request Signals Culling Function**)
- The operation mode of the slave channel must be set to Event Count Mode. (See **Table 23.187, Contents of the TAUDnCMORm register for the Slave Channel of the Interrupt Request Signals Culling Function**)
- TAUDnTTOUTm is not used for the master or slave channel of this function

Functional description

The counters (master and slave) are started by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for both channels. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation. The current value of the data register of the master channel and slave channel (TAUDnCDRm) are written to the counter (TAUDnCNTm).

- Master channel:
When the counter of the master channel reaches 0000_H, INTTAUDnIm is generated and TAUDnCDRm value is reloaded to TAUDnCNTm.
- Slave channel:
Every time the master channel generates an INTTAUDnIm, the counter of the slave channel reduces by one. When the counter reaches 0000_H, it awaits the next interrupt from the master channel. This causes TAUDnCNTm (slave) to reload the value of TAUDnCDRm, and an INTTAUDnIm is generated.

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channel, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm of master and slave channel stop but retain their values.

Conditions

This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.

23.13.9.2 Equations

Interrupt division operator = $TAUDnCDRm$ (slave channel)

- One $INTTAUDnIm$ is generated for the $INTTAUDnIm$ count of the master channel defined by $TAUDnCDRm$ (slave channel) + 1.

23.13.9.3 Block Diagram and General Timing Diagram

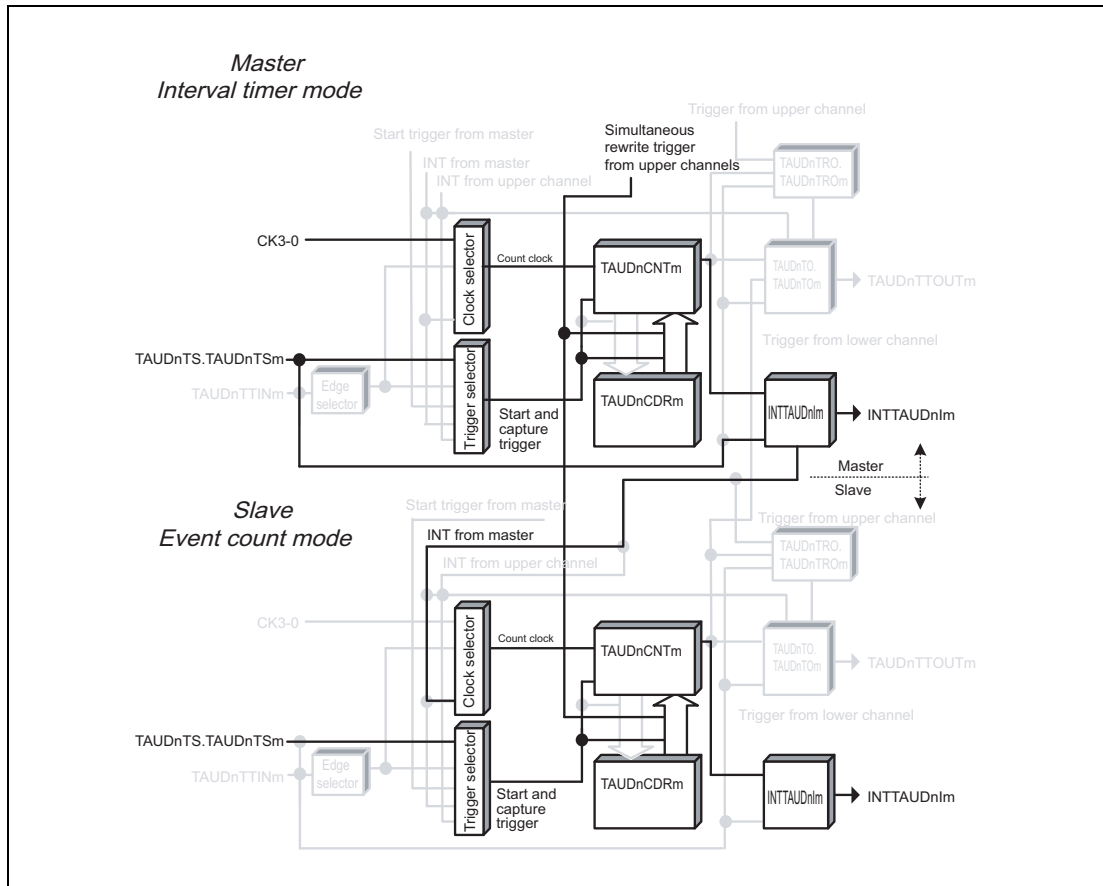


Figure 23.115 Block Diagram of Interrupt Request Signals Culling Function

The following settings apply to the general timing diagram.

Master channel:

- INTTAUDnIm generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

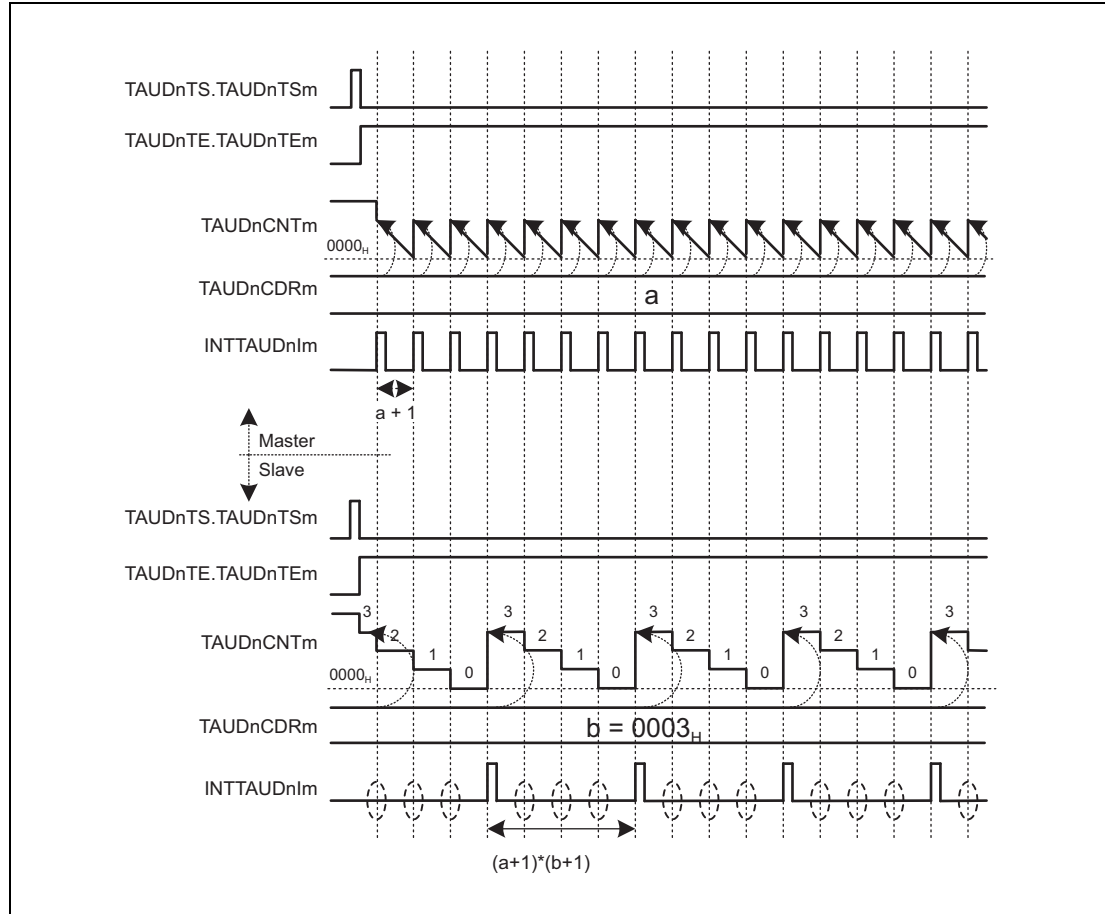


Figure 23.116 General Timing Diagram of Interrupt Request Signals Culling Function

23.13.9.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channels

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.184 Contents of the TAUDnCMORm register for the Master Channel of the Interrupt Request Signals Culling Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation. |

(2) TAUDnCMURm for master channels

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.185 Contents of the TAUDnCMURm register for the Master Channel of the Interrupt Request Signals Culling Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite for the master channel

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.186 Simultaneous Rewrite Settings for the Master channel of Interrupt Request Signals Culling Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.9.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.187 Contents of the TAUDnCMORm register for the Slave Channel of the Interrupt Request Signals Culling Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 11: INTTAUDnIm of the master channel is used as the count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0011: Event count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm not generated at the beginning of operation. |

(2) TAUDnCMURm for slave channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.188 Contents of the TAUDnCMURm register for the Slave Channel of the Interrupt Request Signals Culling Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for the slave channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite for slave channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.189 Simultaneous Rewrite Settings for Slave Channels of Interrupt Request Signals Culling Function

| Bit Name | Setting |
|--------------------|---|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. |

23.13.9.6 Operating Procedure for Interrupt Request Signals Culling Function

Table 23.190 Operating Procedure for Interrupt Request Signals Culling Function

| | Operation | TAUDn Status |
|-------------------------|---|--|
| Initial Channel Setting | Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.9.4, Register Settings for the Master Channel. | Channel operation is stopped. |
| | Slave channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.13.9.5, Register Settings for Slave Channels. | |
| | Set the value of TAUDnCDRm register of every channel. | |
| Restart | Start Operation Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel. |
| | During Operation TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation. | TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000 _H : <ul style="list-style-type: none"> • INTTAUDnIm (master) occurs. • TAUDnCNTm (master) loads TAUDnCDRm value and continues count operation. • TAUDnCNTm of slave channels counts down each time INTTAUDnIm of master channel is detected. When TAUDnCNTm of the slave = 0000 _H : <ul style="list-style-type: none"> • INTTAUDnIm (slave) occurs. • TAUDnCNTm loads the TAUDnCDRm value and continues count operation. |
| Stop Operation | Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retain their current values. |

23.13.9.7 Specific Timing Diagram

(1) Interrupt count (master) = interrupt count (slave)

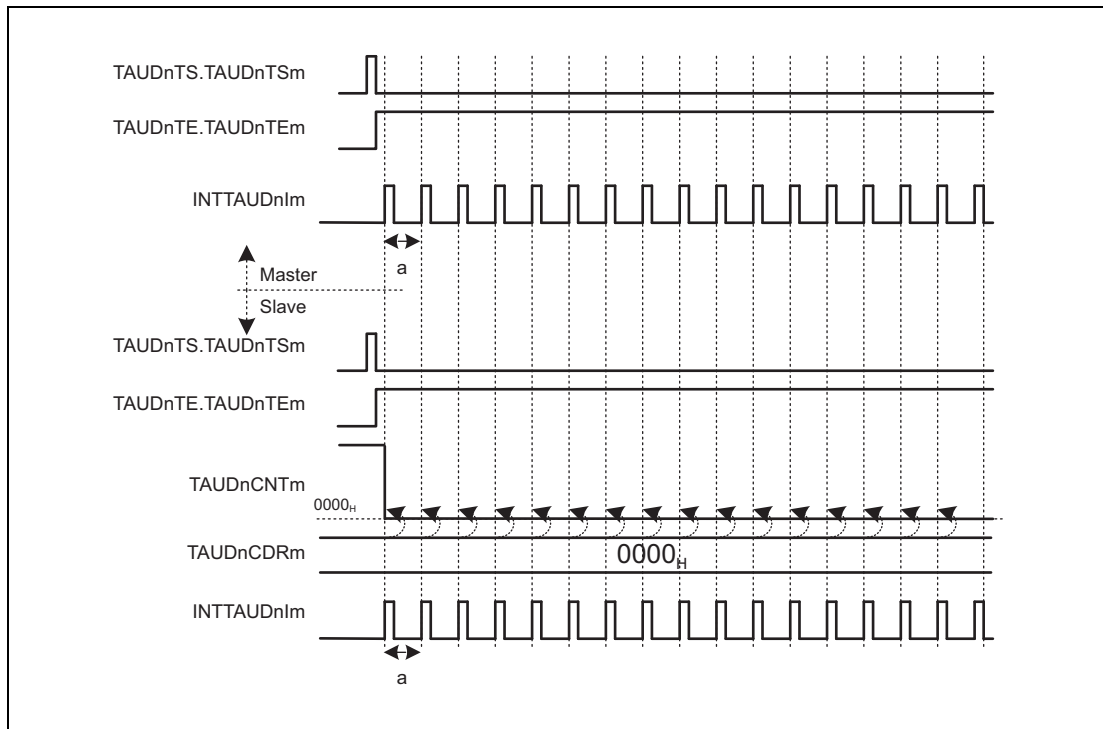


Figure 23.117 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm = 0000_H, TAUDnCDRm value of slave channels is loaded into TAUDnCNTm each time INTTAUDnIm of master channel is detected. In other words, TAUDnCNTm is always 0000_H.
- Therefore, an interrupt occurs on the master channel and simultaneously an interrupt occurs on slave channels.

23.14 Synchronous Non-Complementary and Complementary Modulation Output Functions

This section describes functions that generate 6-phase PWM output or triangle PWM output using a master channel and 7 slave channels.

23.14.1 Non-Complementary Modulation Output Function Type 1

23.14.1.1 Overview

Summary

This function outputs a PWM signal, a high-level signal, or a low-level signal from TAUDnTTOUTm depending on the values of the real-time output bits (TAUDnTRO.TAUDnTROm) and the modulation output enable bits (TAUDnTME.TAUDnTME m) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode. (See **Table 23.192, Contents of the TAUDnCMORm register for the Master Channel of the Non-Complementary Modulation Output Function Type 1**)
- The operating mode for slave channels 1 to 7 should be set to one-count mode. (See **Table 23.195, Contents of the TAUDnCMORm register for the Slave Channel 1 of the Non-Complementary Modulation Output Function Type 1** and **Table 23.199, Contents of the TAUDnCMURm register for the Slave Channels 2 to 7 of the Non-Complementary Output Function Type 1**.)
- TAUDnTTOUTm is not used with the master channel of this function.
- TAUDnTTOUTm of slave channel 1 is not used with this function, but TAUDnTRC.TAUDnTRCm should be set to 1. (See **Section 23.7, Channel Output Modes**)
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 1 with non-complementary modulation output. (See **Section 23.7, Channel Output Modes**)
- TAUDnCDRm of slave channel 1 should be set to 0000_H.

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTE m = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down. When the counter reaches 0000_H, INTTAUDnIm occurs.

- Slave channel 1:
Slave channel 1 is set as a channel that triggers real-time output (TAUDnTRC.TAUDnTRCm = 1). If an interrupt occurs on slave channel 1 (TAUDnCDRm is fixed to 0000_H), the value of real-time output bit (TAUDnTRO.TAUDnTROm) of the channel that monitors the interrupt on slave channel 1 is reflected to TAUDTTOUTm respectively. After that, the counter returns to FFFF_H and waits for the next interrupt of master channel.

- Slave channel 2:
Slave channel 2 generates a PWM output. The master channel specifies a PWM output cycle and slave channel 2 specifies a duty cycle. After generating an interrupt, the counter returns to $FFFF_H$ and awaits the next interrupt from the master channel.

Slave channels 3 to 7 operate like slave channel 2.

As described in **Table 23.191, TAUDnTTOUTm Output from Slave Channels of Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)**, a signal output from TAUDnTTOUTm depends on the value of the real-time output bit (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of slave channel.

This function cannot use a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
 - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs a high-level signal.
 - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
 - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs PWM corresponding to the channel (positive logic).
 - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDnTTOUTm are inverted. PWM signals become negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 23.191 TAUDnTTOUTm Output from Slave Channels of Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)

| TAUDnTME.TAUDnTMEm | TAUDnTRO.TAUDnTROm | TAUDnTTOUTm Output |
|--------------------|--------------------|----------------------|
| 0 | 0 | Low level |
| | 1 | High level |
| 1 | 0 | Low level |
| | 1 | PWM (positive logic) |

- This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.
- TAUDnCDRm value of slave channel 1 should be set to 0000_H so that a real-time output is triggered at the same time as PWM generation on slave channels 2 to 7.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 0 (low level) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 1 (high level) high before TAUDnTE.TAUDnTEm is set to 0.

23.14.1.2 Equations

Slave channels 2 to 7:

Pulse cycle = [TAUDnCDRm (master) + 1] × count clock cycle

Duty time = [TAUDnCDRm (slave)] × count clock cycle

23.14.1.3 Block Diagram and General Timing Diagram

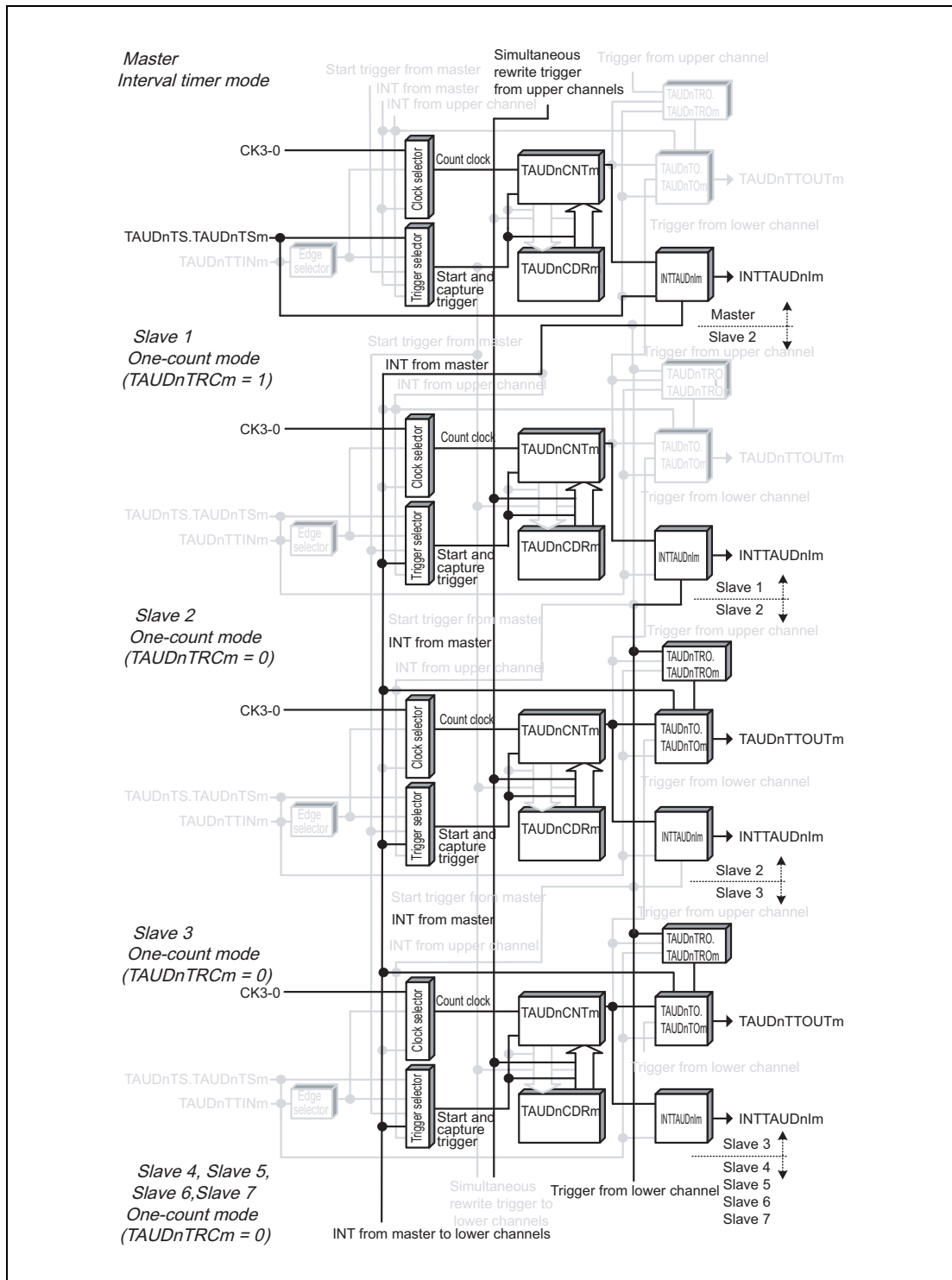


Figure 23.118 Block Diagram of Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

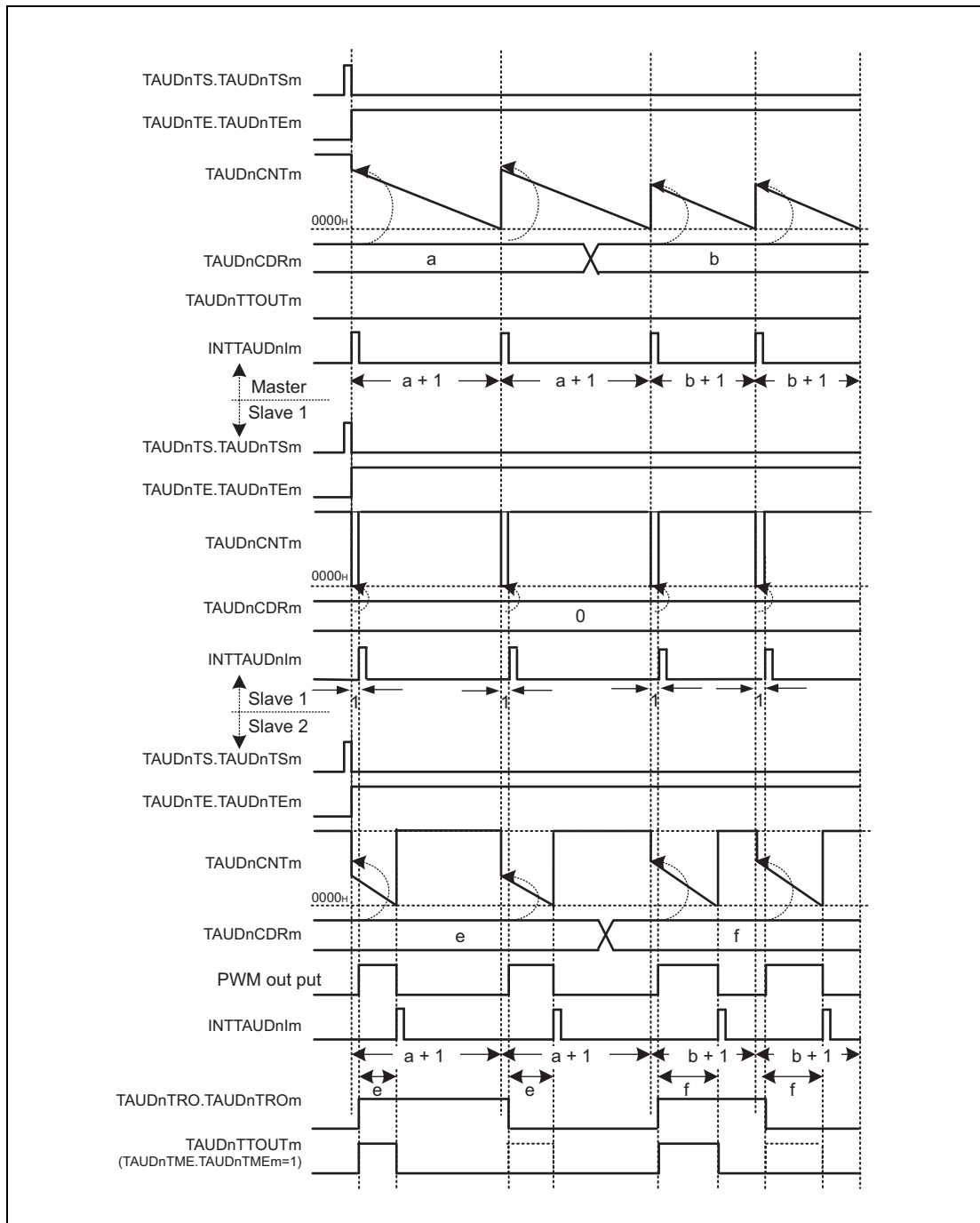


Figure 23.119 General Timing Diagram of Non-Complementary Modulation Output Function Type 1

23.14.1.4 Register Settings for Master Channels

(1) TAUDnCMORm for master channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.192 Contents of the TAUDnCMORm register for the Master Channel of the Non-Complementary Modulation Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 1: INTTAUDnIm is generated at the beginning of operation or at a restart time. |

(2) TAUDnCMURm for master channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.193 Contents of the TAUDnCMURm register for the Master Channel of the Non-Complementary Modulation Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for master channels

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous rewrite for master channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.194 Simultaneous Rewrite Settings for Master Channels of Non-Complementary Modulation Output Function Type 1

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

NOTE

When the TAUDnRDS.TAUDnRDSm bit is 1, operation of a channel with a higher number than the master channel of the type described in Section 23.12.16, Simultaneous Rewrite Trigger Generation Function Type 1, is required. Also make the settings for operation described below.

- Setting the channel for simultaneous rewrite trigger output function type 1: TAUDnRDCm = 1, TAUDnRDSm = 1
The TAUDnCDRm setting for the given channel is derived from the formula below:
= (TAUDnCDRm setting value of master channel for simultaneous rewrite + 1) × interrupt count – 1
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

23.14.1.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for slave channel 1

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKs[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.195 Contents of the TAUDnCMORm register for the Slave Channel 1 of the Non-Complementary Modulation Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKs[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKs[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 100: INTTAUDnIm of master channel is a start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Enables the start trigger during operation |

(2) TAUDnCMURm for slave channel 1

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.196 Contents of the TAUDnCMURm register for the Slave Channel 1 of the Non-Complementary Modulation Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous rewrite for slave channel 1

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.197 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

23.14.1.6 Register Settings for Slave Channels 2 to 7

(1) TAUDnCMORm for slave channels 2 to 7

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.198 Contents of the TAUDnCMORm register for the Slave Channels 2 to 7 of the Non-Complementary Modulation Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 100: INTTAUDnIm of master channel is a start trigger. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Enables the start trigger during operation |

(2) TAUDnCMURm for slave channels 2 to 7

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.199 Contents of the TAUDnCMURm register for the Slave Channels 2 to 7 of the Non-Complementary Output Function Type 1

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for slave channels 2 to 7**Table 23.200 Control Bit Settings in Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROm | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 0: Upper channel generates a real-time output trigger for channel m. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation 1: Enables modulation |

(4) Simultaneous rewrite of slave channels 2 to 7

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.201 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

23.14.1.7 Operating Procedure for Non-Complementary Modulation Output Function Type 1

Table 23.202 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (1/2)

| Operation | TAUDn Status |
|--|-------------------------------|
| <p>Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.1.4, Register Settings for Master Channels.</p> <p>Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.1.5, Register Settings for Slave Channel 1.</p> <p>Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.1.6, Register Settings for Slave Channels 2 to 7.</p> <p>Set the value of TAUDnCDRm register of every channel. Set a pulse cycle with TAUDnCDRm of master channel, 0000_H in TAUDnCDRm of slave channel 1, and duty width with TAUDnCDRm of slave channels 2 to 7.</p> <p>Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.</p> | Channel operation is stopped. |

Initial Channel Setting

Table 23.202 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (2/2)

| | Operation | TAUDn Status |
|---------|--|---|
| Restart | Start Operation Set TAUDnTS.TAUDnTSM of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM of master and slave channels is set to 1 and the counter starts counting down. |
| | During Operation TAUDnCDRm, TAUDnTRO.TAUDnTROM, and TAUDnTME.TAUDnTME can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation. | TAUDnCDRm value of master channel, slave channel 1 and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000 _H : <ul style="list-style-type: none"> • INTTAUDnIm is generated. • TAUDnCDRm value of master channel is reloaded into TAUDnCNTm to continue counting down. • PWM output signals of slave channels 2 to 7 are set/reset. • TAUDnCDRm value of slave channel 1 is reloaded into TAUDnCNTm to perform counting down. • TAUDnCDRm value of slave channels 2 to 7 is reloaded into TAUDnCNTm to perform counting down. • When the counter of slave channel 1 reaches 0000_H: <ul style="list-style-type: none"> – INTTAUDnIm is generated. – TAUDnTRO.TAUDnTROM value of slave channel 2 to 7 is reflected to the TAUDTTOUTm output. • When the counter of slave channels 2 to 7 reaches 0000_H: <ul style="list-style-type: none"> – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set. TAUDnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUDnTRO.TAUDnTROM) and modulation output bit (TAUDnTME.TAUDnTME) of a pair of slave channels. |
| | Stop Operation Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.14.1.8 Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

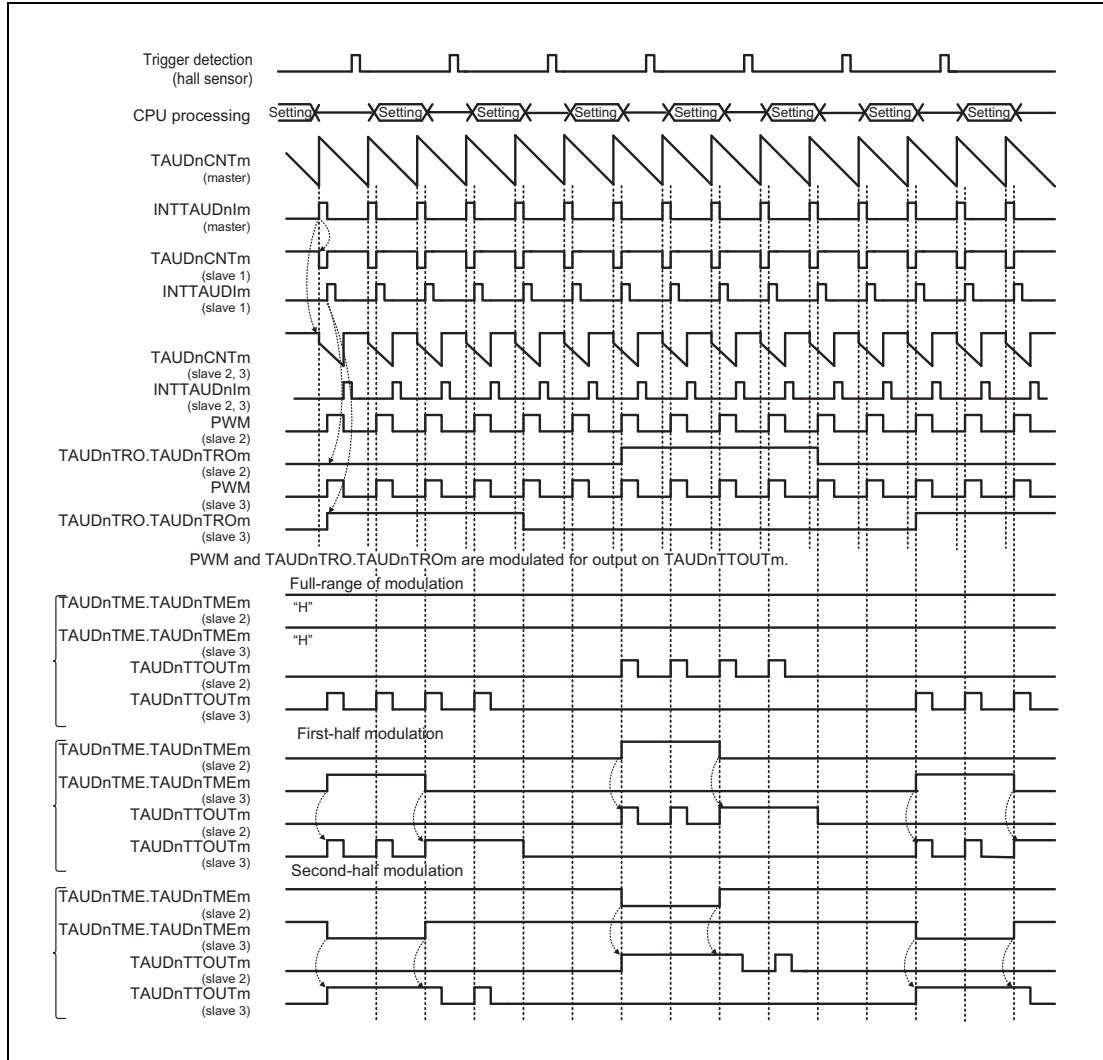


Figure 23.120 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 1

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and master channel cycle. According to the modified setting, modulation waveforms are output from TAUDnTTOUTm.

TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

23.14.2 Non-Complementary Modulation Output Function Type 2

23.14.2.1 Overview

Summary

This function outputs a triangular wave PWM output signal, a high-level signal, or low-level signal from TAUDnTTOUTm depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output enable bit value (TAUDnTME.TAUDnTMEem) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode. (See **Table 23.204, Contents of the TAUDnCMORm register for the Master Channel of the Non-Complementary Modulation Output Function Type 2**)
- The operating mode for slave channel 1 should be set to event count mode. (See **Table 23.208, Contents of the TAUDnCMORm register for the Slave Channel 1 of the Non-Complementary Modulation Output Function Type 2**)
- The operating mode for slave channels 2 to 7 should be set to up/down count mode. (See **Table 23.211, Contents of the TAUDnCMORm register for the Slave Channels 2 to 7 of the Non-Complementary Modulation Output Function Type 2**)
- Output mode of master channel should be set to channel single output mode 1. (See **Section 23.7, Channel Output Modes**)
- This function does not use TAUDnTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1. (See **Section 23.7, Channel Output Modes**)
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with non-complementary modulation output. (See **Section 23.7, Channel Output Modes**)

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm).

- Master channel:
The counter of master channel starts to count down. When the counter reaches 0000_H, INTTAUDnIm occurs.
- Slave channel 1:
When slave channel 1 detects an interrupt from the master channel, the TAUDnCNTm value is decremented by 1. When interrupt count from the master channel reaches TAUDnCDRm + 1, INTTAUDnIm is generated. Afterwards, it leads the TAUDnCDRm value to TAUDnCNTm and continues count operation.
When the counter reaches to 0000_H, slave channel 1 waits from the next interrupt from the master channel. And the TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1) and INTTAUDnIm is generated.
Since slave channel 1 is set as a real-time output trigger channel

(TAUDnTRC.TAUDnTRCm = 1), the real-time output bit (TAUDnTRO.TAUDnTROm) of the channel which monitors an interrupt on the corresponding channel is reflected to respective TAUDTTOUTm output when an interrupt occurs on slave 1 channel.

- Slave channel 2:
Once an interrupt is detected from the master channel, TAUDnCNTm counts in the reverse direction. When an interrupt is detected during count-up operation, TAUDnCDRm value is reloaded and then the counter starts to count down.
If TAUDnCNTm = 0001_H, an interrupt occurs and a PWM output signal is set/reset.

The combined use of the master channel and slave channel 2 generates a PWM output signal. The master channel generates a PWM output cycle and slave channel 2 generate a duty cycle.

Slave channels 3 to 7 operates like slave channel 2.

A signal that is output from TAUDnTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm) and a modulation output bit value (TAUDnTME.TAUDnTME_m) of the slave channel, as described in **Table 23.203, TAUDnTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)**.

This function cannot make a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTE_m to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTME_m = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
 - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs PWM corresponding to the channel (positive logic).
 - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTME_m = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
 - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs a high-level signal.
 - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDnTTOUTm are inverted. PWM signals become negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 23.203 TAUDnTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)

| TAUDnTME.TAUDnTME _m | TAUDnTRO.TAUDnTROm | TAUDnTTOUTm Output |
|--------------------------------|--------------------|----------------------|
| 0 | 0 | Low level |
| | 1 | High level |
| 1 | 0 | Low level |
| | 1 | PWM (positive logic) |

- This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOM is set 0 (low level) before TAUDnTE.TAUDnTEM is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOM is set 1 (high level) before TAUDnTE.TAUDnTEM is set to 0.

23.14.2.2 Equations

Slave channels 2 to 7:

Carrier cycle (down/up)

$$= [\text{TAUDnCDRm (master)} + 1] \times 2 \times \text{count clock cycle}$$

Duty time

$$= [\text{TAUDnCDRm (master)} + 1 - \text{TAUDnCDRm (slave)}] \times 2 \times \text{count clock cycle}$$

23.14.2.3 Block Diagram and General Timing Diagram

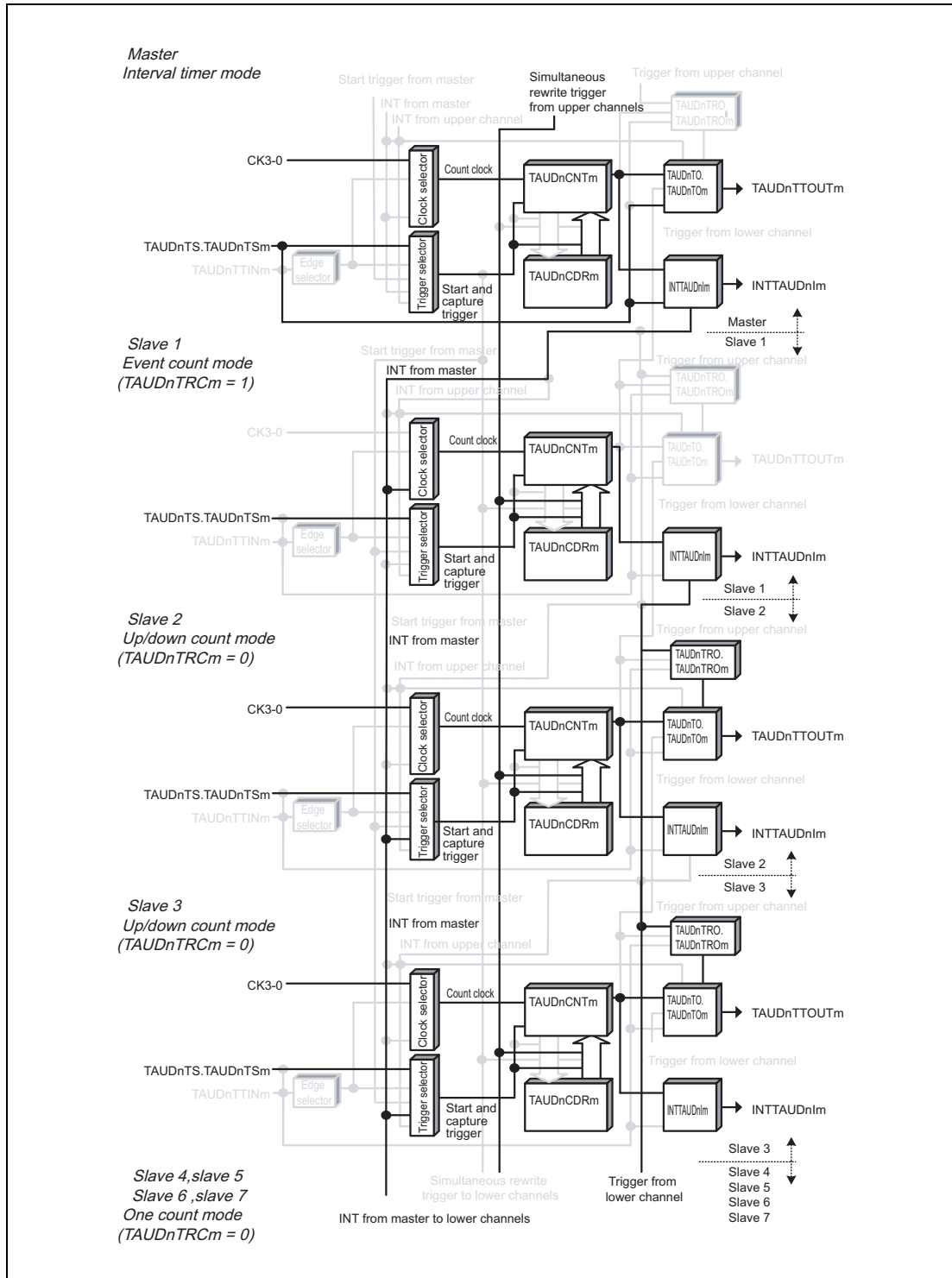


Figure 23.121 Block Diagram of Non-Complementary Modulation Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

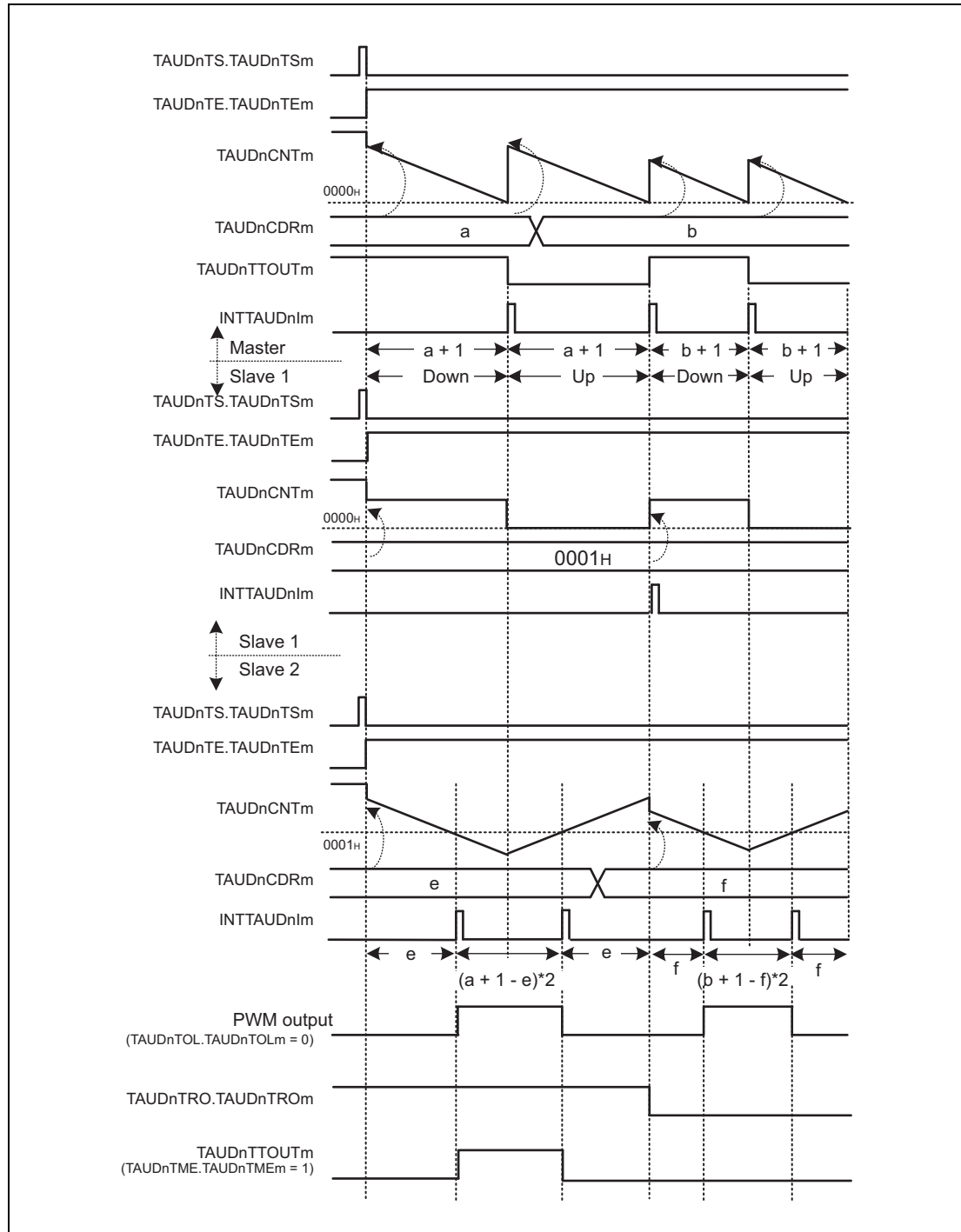


Figure 23.122 General Timing Diagram of Non-Complementary Modulation Output Function Type 2

23.14.2.4 Register Settings for Master Channels

(1) TAUDnCMORm for master channels

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKs[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.204 Contents of the TAUDnCMORm register for the Master Channel of the Non-Complementary Modulation Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKs[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKs[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated at the beginning of operation or at a restart time. |

(2) TAUDnCMURm for master channels

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.205 Contents of the TAUDnCMURm register for the Master Channel of the Non-Complementary Modulation Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode for master channels**Table 23.206 Control Bit Settings for Master Channels in Non-Complementary Modulation Output Function Type 2**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (toggle mode with TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: Setting is disabled (value after reset) in toggle mode. |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: Set 0 while real-time output is prohibited (TAUDnTRE.TAUDnTREm = 0). |
| TAUDnTRC.TAUDnTRCm | 0: Set 0 while real-time output is prohibited (TAUDnTRE.TAUDnTREm = 0). |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for master channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.207 Simultaneous Rewrite Settings for Master Channels of Non-Complementary Modulation Output Function Type 2

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

23.14.2.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for slave channel 1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.208 Contents of the TAUDnCMORm register for the Slave Channel 1 of the Non-Complementary Modulation Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 11: INTTAUDnIm of the master channel is used as the count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. 011: Triggers simultaneous rewrite. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0011: Event count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. |

(2) TAUDnCMURm for slave channel 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R | R/W |

Table 23.209 Contents of the TAUDnCMURm register for the Slave Channel 1 of the Non-Complementary Modulation Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous rewrite for slave channel 1

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.210 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

23.14.2.6 Register Settings for slave channels 2 to 7

(1) TAUDnCMORm for slave channels 2 to 7

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.211 Contents of the TAUDnCMORm register for the Slave Channels 2 to 7 of the Non-Complementary Modulation Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 111: The up/down output trigger signal of the master channel |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1001: Up/down count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. |

(2) TAUDnCMURm for slave channels 2 to 7

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.212 Contents of the TAUDnCMURm register for the Slave Channels 2 to 7 of the Non-Complementary Modulation Output Function Type 2

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Output mode for slave channels 2 to 7**Table 23.213 Control Bit Settings in Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROm | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 0: The upper channel generates the real-time output trigger for channel m |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation 1: Enables modulation |

(4) Simultaneous rewrite for slave channels 2 to 7

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.214 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

23.14.2.7 Operating Procedure for Non-Complementary Modulation Output Function Type 2

Table 23.215 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (1/2)

| Operation | TAUDn Status |
|---|-------------------------------|
| <p>Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.2.4, Register Settings for Master Channels.</p> <p>Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.2.5, Register Settings for Slave Channel 1.</p> <p>Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.2.6, Register Settings for slave channels 2 to 7.</p> <p>Set the value of TAUDnCDRm register of every channel. Set pulse cycle in TAUDnCDRm of master channel, and in TAUDnCDRm of slave channel 1, set the number of interrupts from master channel to be ignored before slave channel 1 generates a real-time output trigger. Set duty width in TAUDnCDRm of slave channels 2 to 7.</p> <p>Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.</p> | Channel operation is stopped. |

Initial Channel Setting

Table 23.215 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (2/2)

| | Operation | TAUDn Status |
|---------|--|---|
| Restart | Start Operation Set TAUDnTS.TAUDnTSM of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSM is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM of master and slave channels is set to 1 and the counter starts counting down. |
| | During Operation TAUDnCDRm, TAUDnTRO.TAUDnTROM, and TAUDnTME.TAUDnTME can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation. | <p>TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform count down. The TAUDnCDRm value of slave channel 1 is loaded and the counter waits for an interrupt from the master channel. When the counter of master channel reaches 0000_H:</p> <ul style="list-style-type: none"> • INTTAUDnIm is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. • The TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for a next interrupt from the master channel. • TAUDnCNTm of slave channels 2 to 7 reloads the TAUDnCDRm value or starts counting in opposite direction. • When the TAUDnCDRm value is loaded, the TAUDnTRO.TAUDnTROM value of slave channel 2 to 7 is reflected to the TAUDTTOUTm output. • When slave channel detects an interrupt of the master channel for TAUDnCDRm + 1 count. <ul style="list-style-type: none"> – INTTAUDnIm is generated. – The TAUDnTRO.TAUDnTROM value of slave channel 2 to 7 is reflected to the TAUDTTOUTm output. • When the counter of slave channel 1 reaches 0000_H, it waits for a next interrupt from the master channel. When an interrupt is detected: <ul style="list-style-type: none"> – INTTAUDnIm is generated. • When the counter of slave channels 2 to 7 reaches 0001_H: <ul style="list-style-type: none"> – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set/reset. <p>TAUDnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUDnTRO.TAUDnTROM) and modulation output bit (TAUDnTME.TAUDnTME) of a pair of slave channels.</p> |
| | Stop Operation Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0. | TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values. |

23.14.2.8 Specific Timing Diagrams

The following settings apply to the general timing diagram.

- Master channel: No INTTAUDnIm is generated at start of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

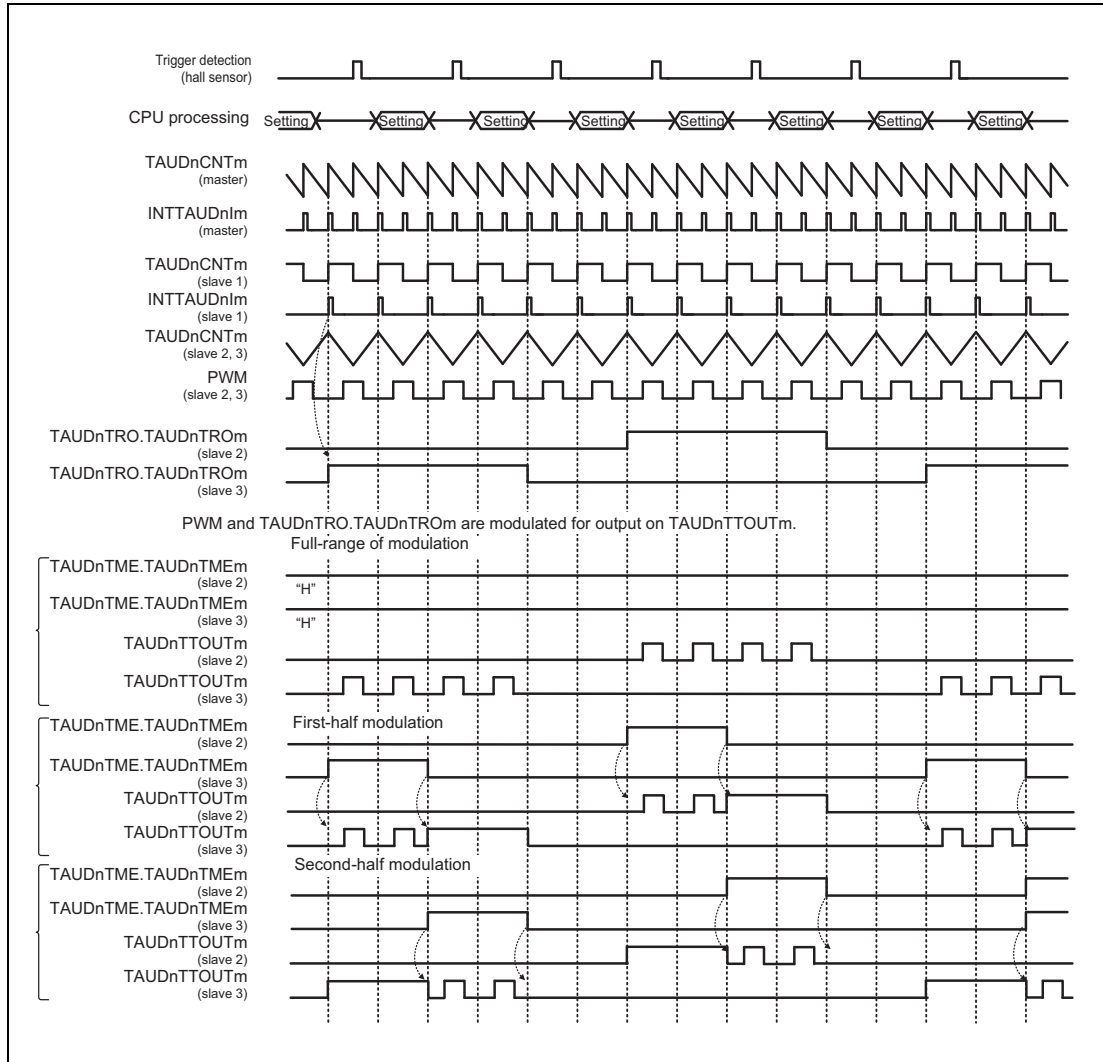


Figure 23.123 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 2

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

23.14.3 Complementary Modulation Output Function

23.14.3.1 Overview

Summary

This function outputs a triangular PWM output signal, a high-level signal, or low-level signal from TAUDnTTOUTm adding dead time depending on the real-time output bit value (TAUDnTRO.TAUDnTROm), the modulation output bit value (TAUDnTME.TAUDnTME m), and the output level bit value (TAUDnTDL.TAUDnTDLm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operating mode of the master channel must be set to interval timer mode. (See **Table 23.217, Contents of the TAUDnCMORm register for the Master Channel of the Complementary Modulation Output Function**)
- The operating mode for slave channel 1 should be set to event count mode. (See **Table 23.221, Contents of the TAUDnCMORm register for the Slave Channel 1 of the Complementary Modulation Output Function**)
- The operating mode for slave channels 2, 4 and 6 should be set to up/down count mode. (See **Table 23.224, Contents of the TAUDnCMORm register for the Slave Channels 2, 4, and 6 of the Complementary Modulation Output Function**)
- The operating mode for slave channels 3, 5 and 7 should be set to one-count mode. (See **Table 23.228, Contents of the TAUDnCMORm register for the Slave Channels 3, 5, and 7 of the Complementary Modulation Output Function**). In addition, as the number of occurrences of an interrupt for slave channels 3, 5 and 7 within the carrier cycle is not uniquely determined, do not use the interrupt as an interrupt source.
- The output mode for master channels should be set to independent channel output mode 1. (See **23.7, Channel Output Modes**)
- This function does not use TAUDnTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1. (See **Section 23.7, Channel Output Modes**)
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with complementary modulation output. (See **Section 23.7, Channel Output Modes**)

Functional description

- Master channel:
The counter of master channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSM) to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The value of data register (TAUDnCDRm) of master channel is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.
When the counter of master channel reaches 0000_H, INTTAUDnIm occurs. This decrements the counter value of slave channel 1 by 1 and the counter of slave channel 2 starts to count in the opposite direction.

- Slave channel 1:
When the counter reaches 0000_H , slave channel 1 waits for the next interrupt from the master channel. And the $TAUDnCDRm$ value is reloaded into $TAUDnCNTm$ (slave 1) and $INTTAUDnIm$ is generated.
Slave channel 1 is set as a real-time output trigger channel ($TAUDnTRC.TAUDnTRCm = 1$). The value of real-time output bit ($TAUDnTRO.TAUDnTROM$) of each channel is applied to the channel that detects the occurrence of an interrupt on slave channel 1 by an interrupt. The real-time output bit value can be changed in any timing by application software but a new value is not applied until an interrupt occurs on slave channel 1.
- Slave channel 2:
When the slave channel 2 counter reaches 0001_H , the slave channel 3 counter starts counting down. When the slave channel 3 counter reaches 0000_H , an interrupt occurs.
- Slave channels 2 and 3:
The combined use of the master channel and slave channels 2 and 3 generates a PWM output signal. The master channel generates a PWM output cycle, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time.
- Slave channels 4 to 7:
Slave channels 4 and 6 operate like slave channel 2. Slave channels 5 and 7 operate like slave channel 3.

As in **Table 23.216, TAUDnTTOUTm Output ($TAUDnTOL.TAUDnTOLm = 0$) for a Pair of Slave Channels of Complementary Modulation Output Function**, a signal that is output from $TAUDnTTOUTm$ depends on a real-time output bit value ($TAUDnTRO.TAUDnTROM$), a modulation output bit value ($TAUDnTME.TAUDnTME m$), and an output level bit value ($TAUDnTDL.TAUDnTDLm$) of the slave channel.

It is, however, prohibited that a high-level signal is output from both channel 2 and channel 3 (in order to prevent a motor driver short circuit). This function cannot perform a forcible restart.

The counter can be stopped by setting $TAUDnTT.TAUDnTTm$ of master and slave channels to 1. This sets $TAUDnTE.TAUDnTE m$ to 0. $TAUDnCNTm$ and $TAUDnTTOUTm$ of master and slave channels stop but retain their values. The counters can be restarted by setting $TAUDnTS.TAUDnTSm$ to 1.

Conditions

- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 1 (TAUDnTOL.TAUDnTOLm = 0):
 - If TAUDnTRO.TAUDnTROm of one channel is set to 1, TAUDnTTOUTm outputs the corresponding PWM of the channel.
 - If TAUDnTRO.TAUDnTROm of each channel is set to 0, a pair of TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 0 (TAUDnTOL.TAUDnTOLm = 0):
 - If TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm of the channel outputs a high-level signal.
 - If TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm of the channel outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDnTTOUTm are inverted. PWM signals become negative logic.

Table 23.216 TAUDnTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function

| TAUDnTME. TAUDnTME2 | TAUDnTME. TAUDnTME3 | TAUDnTRO. TAUDnTRO2 | TAUDnTRO. TAUDnTRO3 | TAUDnTDL. TAUDnTDL2 | TAUDnTDL. TAUDnTDL3 | TAUDnTTOUT2 Output | TAUDnTTOUT3 Output |
|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|-----------------------|-----------------------|
| 0 | 0 | 0 | 0 | X | X | Low level | Low level |
| | | 0 | 1 | 0 | 1 | Low level | High level |
| | | 1 | 0 | 1 | 0 | High level | Low level |
| | | 1 | 1 | X | X | Setting prohibited | Setting prohibited |
| 1 | 1 | 0 | 0 | X | X | Low level | Low level |
| | | 0 | 1 | 0 | 1 | ~PWMm | PWMm |
| | | 1 | 0 | 1 | 0 | PWMm | ~PWMm |
| | | 1 | 1 | X | X | Setting prohibited | Setting prohibited |

NOTES

1. PWM in this Table indicates a normal PWM signal and ~PWM indicates an inverted PWM signal (positive logic). PWM and ~PWM are set by TAUDnTDL.TAUDnTDLm.
2. Any settings not listed above are prohibited.

- If TAUDnTME.TAUDnTMEm is continuously set to 1 while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, full modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the first half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, first-half modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the second half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, second-half modulation is applied.

- Whether dead time is added to a normal or inverted phase PWM signal when two channels become high-level signal outputs simultaneously depends on a TAUDnTDL.TAUDnTDLm bit value.
 - If TAUDnTDL.TAUDnTDLm = 0, dead time is added to a normal phase PWM signal.
 - If TAUDnTDL.TAUDnTDLm = 1, dead time is added to a reverse phase PWM signal.
 - The operation defined by a TAUDnTDL.TAUDnTDLm bit value should be conducted by application software during operation. To modify TAUDnTDL.TAUDnTDLm, rewrite it during the period when TAUDnTRO.TAUDnTROm is 00_B.
- The value of TAUDnCDRm of slave channel 1 should be set to the value that generates INTTAUDnIm on slave channel 1 at the peak of a carrier cycle.
- Set TAUDnCMORm.TAUDnMD0 of master channel to 0.
- This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.
- When TAUDnTOL.TAUDnTOLm = 0 is set in slave channel 2 to 7:
 - If TAUDnTDL.TAUDnTDLm = 0, set TAUDnTO.TAUDnTOm to 0 (low level) before setting TAUDnTE.TAUDnTEm = 0.
 - If TAUDnTDL.TAUDnTDLm = 1, set TAUDnTO.TAUDnTOm to 1 (high level) before setting TAUDnTE.TAUDnTEm = 0.
- When TAUDnTOL.TAUDnTOLm = 1 is set in slave channel 2 to 7:
 - If TAUDnTDL.TAUDnTDLm = 0, set TAUDnTO.TAUDnTOm to 1 (high level) before setting TAUDnTE.TAUDnTEm = 0.
 - If TAUDnTDL.TAUDnTDLm = 1, set TAUDnTO.TAUDnTOm to 0 (low level) before setting TAUDnTE.TAUDnTEm = 0.

23.14.3.2 Equations

Pulse cycle = [TAUDnCDRm (master) + 1] × count clock cycle

$0000_H \leq \text{TAUDnCDRm (master)} < \text{FFFF}_H$

Carrier cycle (down/up) = [TAUDnCDRm (master) + 1] × 2 × count clock cycle

For slave channel 2 and 3:

PWM signal width (normal) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (inverted) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

For slave channels 4 to 7:

Calculate slave channel 4 and 6 in the same way as slave channel 2 whereas calculate slave channel 5 and 7 in the same way as slave channel 3.

23.14.3.3 Block Diagram and General Timing Diagram

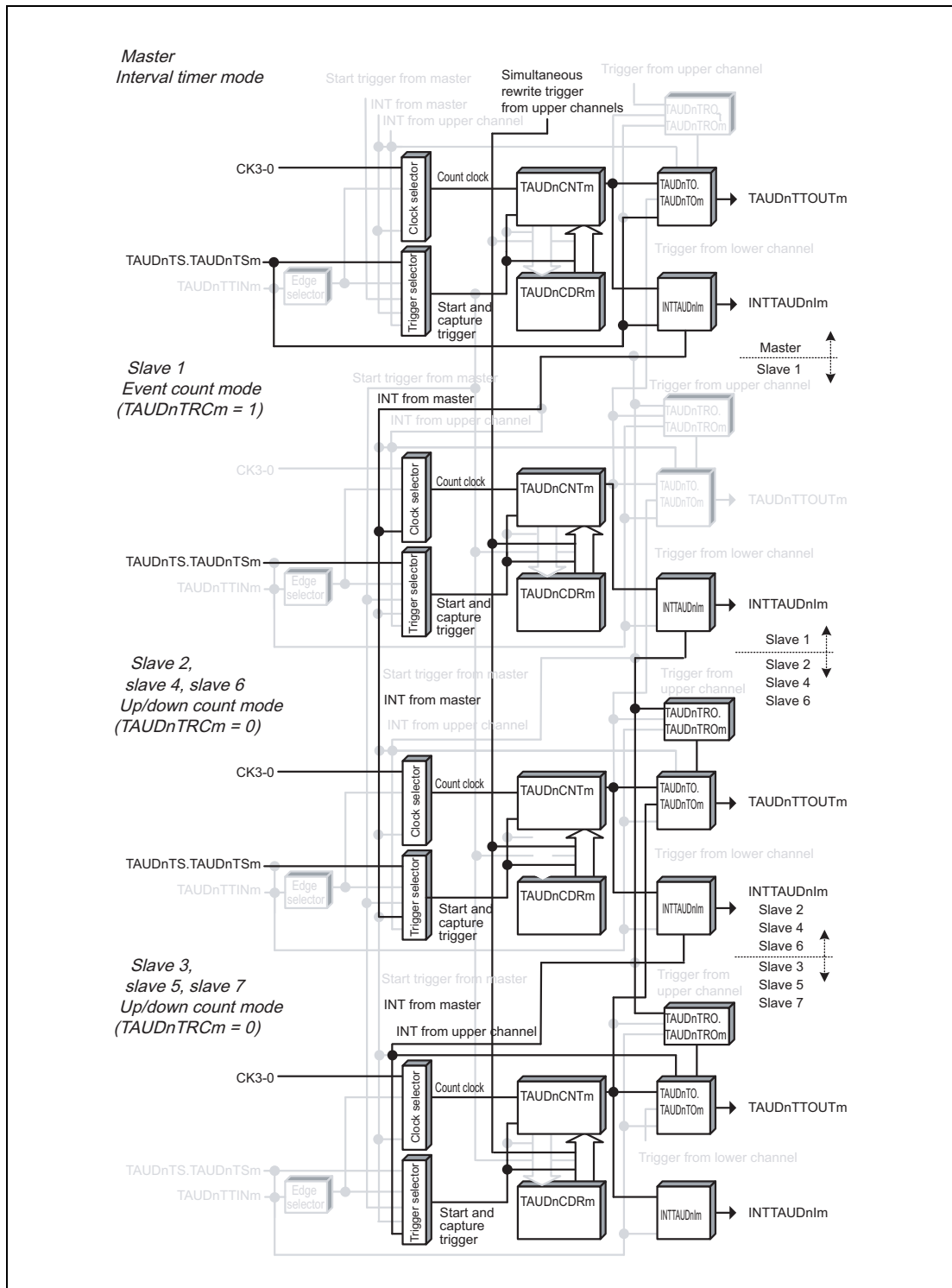


Figure 23.124 Block Diagram of Complementary Modulation Output Function

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 1: TAUDnCDRm=0001_H

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

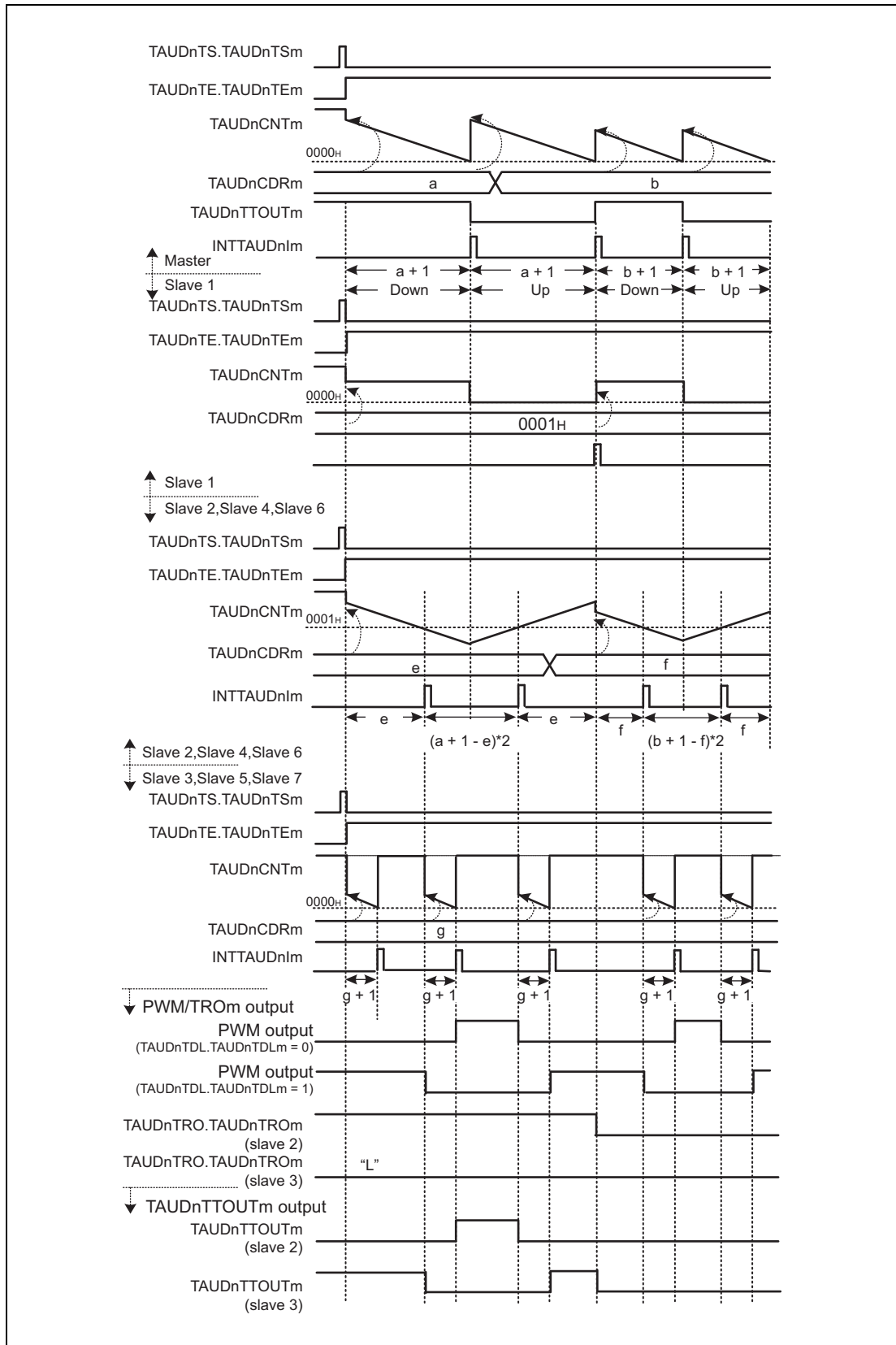


Figure 23.125 General Timing Diagram of Complementary Modulation Output Function

23.14.3.4 Register Settings for Master Channels

(1) TAUDnCMORm for master channels

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.217 Contents of the TAUDnCMORm register for the Master Channel of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 1: Channel is master channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0000: Interval timer mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated and TAUDnTTOUTm is not toggled at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated to toggle TAUDnTTOUTm at the beginning of an operation. |

(2) TAUDnCMURm for master channels

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.218 Contents of the TAUDnCMURm register for the Master Channel of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode of master channel**Table 23.219 Control Bit Settings in Independent Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 0: Independent channel output |
| TAUDnTOC.TAUDnTOCm | 0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0) |
| TAUDnTOL.TAUDnTOLm | 0: Setting is disabled (value after reset) in toggle mode. |
| TAUDnTDE.TAUDnTDEm | 0: Disables dead time operation |
| TAUDnTDM.TAUDnTDMm | 0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0 |
| TAUDnTDL.TAUDnTDLm | |
| TAUDnTRE.TAUDnTREm | 0: Disables real-time output |
| TAUDnTRO.TAUDnTROm | 0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0 |
| TAUDnTRC.TAUDnTRCm | |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation |

(4) Simultaneous rewrite for master channels

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.220 Simultaneous Rewrite Settings for Master Channels of Complementary Modulation Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than a master channel to generate a simultaneous rewrite trigger signal.

23.14.3.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for slave channel 1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.221 Contents of the TAUDnCMORm register for the Slave Channel 1 of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 11: INTTAUDnIm of the master channel is used as the count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 000: Trigger the counter using software. 011: Triggers simultaneous rewrite. |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0011: Event count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. |

(2) TAUDnCMURm for slave channel 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.222 Contents of the TAUDnCMURm register for the Slave Channel 1 of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous rewrite for slave channel 1

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.223 Simultaneous Rewrite Settings for Slave Channel 1 of Complementary Modulation Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

23.14.3.6 Register Settings for slave channels 2, 4, and 6

(1) TAUDnCMORm for slave channels 2, 4, and 6

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.224 Contents of the TAUDnCMORm register for the Slave Channels 2, 4, and 6 of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUDnCKS[1:0] bit of the master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 111: Up/down output trigger signal of master channel |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 1001: Up/down count mode |
| 0 | TAUDnMD0 | 0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. |

(2) TAUDnCMURm for slave channels 2, 4, and 6

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.225 Contents of the TAUDnCMURm register for the Slave Channels 2, 4, and 6 of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Output mode for slave channels 2, 4, and 6**Table 23.226 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 1: Enables dead time operation. |
| TAUDnTDM.TAUDnTDMm | 0: Adds dead time upon detection of a duty cycle on an even-numbered upper channel and when the condition set in TAUDnTDL.TAUDnTDLm is met. |
| TAUDnTDL.TAUDnTDLm | 0: Adds dead time to normal phase. 1: Adds dead time to reverse phase. |
| TAUDnTRE.TAUDnTREM | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROM | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 0: Upper channel generates a real-time output trigger for channel m. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation 1: Enables modulation |

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from odd-numbered channels during PWM output.

(4) Simultaneous rewrite for slave channels 2, 4, and 6

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.227 Simultaneous Rewrite Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDSm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDSm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

23.14.3.7 Register Settings for slave channels 3, 5, and 7

(1) TAUDnCMORM for slave channels 3, 5, and 7

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUDnCKS[1:0] | | TAUDnCCS[1:0] | | TAUDnMAS | TAUDnSTS[2:0] | | TAUDnCOS[1:0] | | — | TAUDnMD[4:1] | | | | TAUDnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 23.228 Contents of the TAUDnCMORM register for the Slave Channels 3, 5, and 7 of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUDnCKS[1:0] | Operating Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The TAUDnCKS[1:0] values of master and slave channels must be identical. |
| 13, 12 | TAUDnCCS[1:0] | 00: Uses an operating clock as a count clock |
| 11 | TAUDnMAS | 0: Slave channel |
| 10 to 8 | TAUDnSTS[2:0] | 110: Dead time trigger |
| 7, 6 | TAUDnCOS[1:0] | 00: Unused. Set to 00. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUDnMD[4:1] | 0100: One-count mode |
| 0 | TAUDnMD0 | 1: Enables start trigger detection while counting. |

(2) TAUDnCMURM for slave channels 3, 5, and 7

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUDnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 23.229 Contents of the TAUDnCMURM register for the Slave Channels 3, 5, and 7 of the Complementary Modulation Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUDnTIS[1:0] | 00: Unused. Set to 00. |

(3) Output mode for slave channels 3, 5, and 7**Table 23.230 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output**

| Bit Name | Setting |
|--------------------|---|
| TAUDnTOE.TAUDnTOEm | 1: Enables independent channel output mode |
| TAUDnTOM.TAUDnTOMm | 1: Synchronous channel output |
| TAUDnTOC.TAUDnTOCm | 1: Operating mode 2 |
| TAUDnTOL.TAUDnTOLm | 0: Positive logic 1: Negative logic |
| TAUDnTDE.TAUDnTDEm | 1: Enables dead time operation. |
| TAUDnTDM.TAUDnTDMm | 0: Adds dead time upon detection of a duty cycle on an even-numbered upper channel and when the condition set in TAUDnTDL.TAUDnTDLm is met. |
| TAUDnTDL.TAUDnTDLm | 0: Adds dead time to normal phase. 1: Adds dead time to reverse phase. |
| TAUDnTRE.TAUDnTREM | 1: Enables real-time output. |
| TAUDnTRO.TAUDnTROM | 0: Real-time output is low. 1: Real-time output is high. |
| TAUDnTRC.TAUDnTRCm | 0: Upper channel generates a real-time output trigger for channel m. |
| TAUDnTME.TAUDnTMEm | 0: Disables modulation 1: Enables modulation |

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from even-numbered channels during PWM output.

(4) Simultaneous rewrite for slave channels 3, 5, and 7

Both master and slave channels should have the same simultaneous rewrite settings.

Table 23.231 Simultaneous Rewrite Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function

| Bit Name | Setting |
|--------------------|--|
| TAUDnRDE.TAUDnRDEm | 1: Enables simultaneous rewrite. |
| TAUDnRDS.TAUDnRDsm | 0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers. |
| TAUDnRDM.TAUDnRDMm | 1: A simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave. |
| TAUDnRDC.TAUDnRDCm | 0: Does not operate as a simultaneous rewrite trigger generation channel. When TAUDnRDS.TAUDnRDsm = 0, the master channel is monitored for simultaneous rewrite triggers regardless of the value of this bit. |

23.14.3.8 Operating Procedure for Complementary Modulation Output Function

Table 23.232 Operating Procedure for Complementary Modulation Output Function (1/2)

| Operation | TAUDn Status |
|--|--|
| <p>Master channels: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.3.4, Register Settings for Master Channels.</p> <p>Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.3.5, Register Settings for Slave Channel 1.</p> <p>Slave channels 2, 4, and 6: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.3.6, Register Settings for slave channels 2, 4, and 6.</p> <p>Slave channels 3, 5, and 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in Section 23.14.3.7, Register Settings for slave channels 3, 5, and 7.</p> <p>Set the value of TAUDnCDRm register of every channel. Set a pulse cycle using TAUDnCDRm of master channel, and an interrupt count of master channel ignored using TAUDnCDRm of slave channel 1. Also set a duty width in TAUDnCDRm of slave channels 2, 4, and 6, and a dead time delay on slave channels 3, 5, and 7.</p> <p>Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.</p> | <p>Channel operation is stopped.</p> |
| <p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p> | <p>TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.</p> |

Initial Channel Setting

Restart

Table 23.232 Operating Procedure for Complementary Modulation Output Function (2/2)

| | Operation | TAUDn Status |
|------------------|---|--|
| Restart | <p>TAUDnCDRm, TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEEm, and TAUDnTDL.TAUDnTDLm can be changed at any time.</p> <p>TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p> | <p>TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform count down. TAUDnCDRm value of slave channel 1 is loaded and the counter waits for a master channel interrupt. When the counter of master channel reaches 0000_H:</p> <ul style="list-style-type: none"> • INTTAUDnIm is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. • TAUDnCNTm value of slave channel 1 decreases by 1 and the counter waits for the next master channel interrupt. • TAUDnCNTm of slave channels 2, 4, and 6 reloads the TAUDnCDRm value or performs counting in the reverse direction. • When the TAUDnCDRm values of slave channel 2, 4, and 6 are loaded, the TAUDnTME.TAUDnTMEEm of slave 2 to 7 are reflected to the TAUDTTOUTm output. • The counter of slave channel 1 waits for the next interrupt from the master channel when reaching 0000_H. When the interrupt is detected: <ul style="list-style-type: none"> – TAUDnCDRm value is reloaded into TAUDnCNTm and the counter waits for the next master channel interrupt. – INTTAUDnIm is generated. – TAUDnTRO.TAUDnTROm is changeable. • When the counter of slave channels 2, 4, and 6 reaches 0001_H: <ul style="list-style-type: none"> – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of channel output mode is met). – TAUDnCDRm value of slave channels 3, 5, and 7 is loaded into TAUDnCNTm to perform count down. • When the counter of slave channels 3, 5, and 7 reaches 0000_H: <ul style="list-style-type: none"> – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of channel output mode is met). <p>TAUDnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bit (TAUDnTRO.TAUDnTROm), modulation output bit (TAUDnTME.TAUDnTMEEm), and output level bit (TAUDnTDL.TAUDnTDLm) of a pair of slave channels.</p> |
| During Operation | | |
| Stop Operation | <p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously.</p> <p>TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p> | <p>TAUDnTE.TAUDnTEEm is cleared to 0 and the counter stops.</p> <p>TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.</p> |

23.14.3.9 Specific Timing Diagrams

The following settings apply to the timing diagram.

- Master channel: INTTAUDnIm is not generated at startup (TAUDnCMORm.TAUDnMD0 = 0).
- Slave channel 1: TAUDnCDRm=0001_H
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

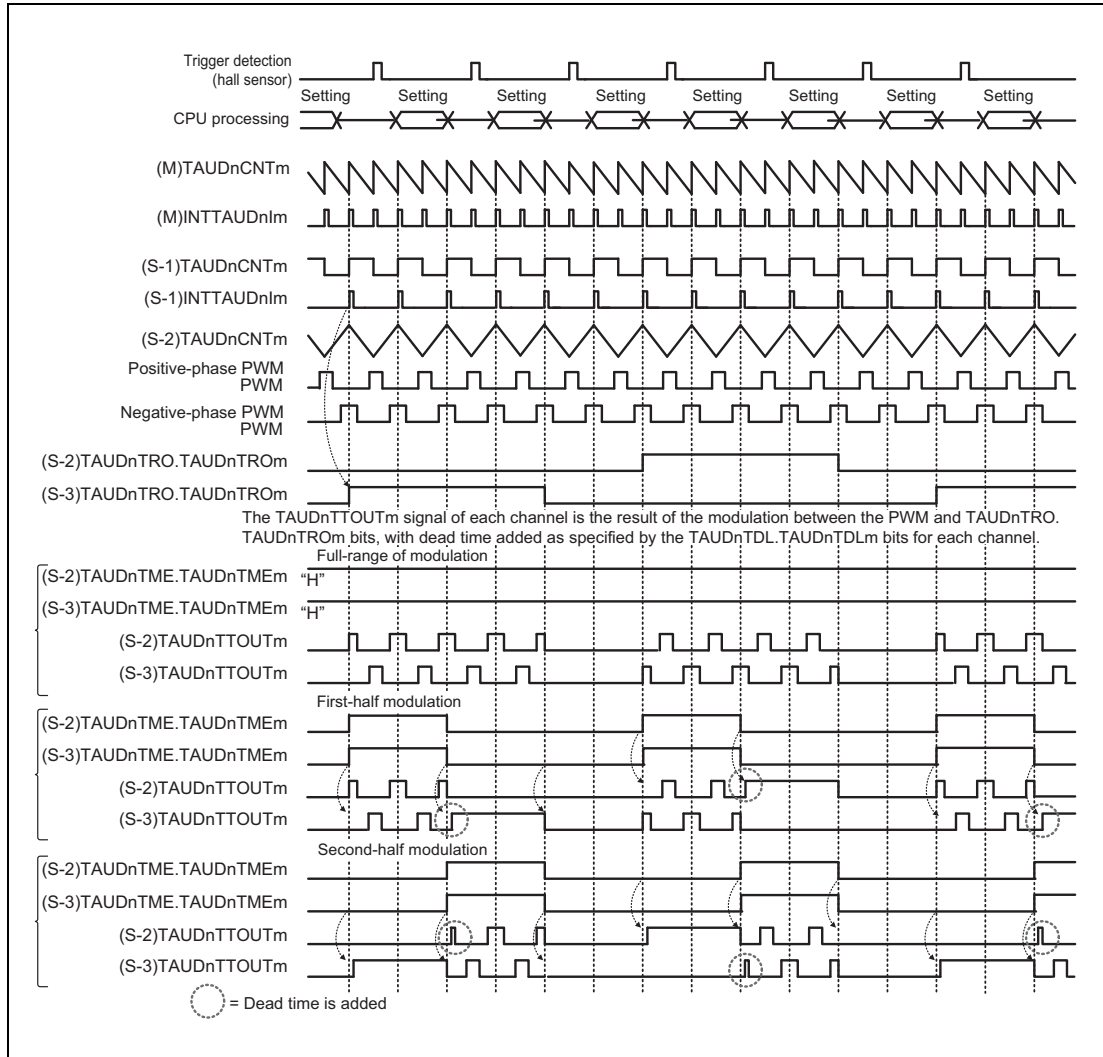


Figure 23.126 Specific Timing Diagram of Complementary Modulation Output Function

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

A modulated PWM output signal and TAUDnTRO.TAUDnTROm bit value are output from slave channels 2 and 3. Therefore, a type (normal/reverse phase) of PWM signal to be output from each channel depends on this bit value. Setting values of TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are reflected at count start timing and upon a carrier period (peak) of a triangular wave PWM.

TAUDnTRO.TAUDnTROm bit value is specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

NOTE

Dead time is added to suppress simultaneous change of PWM edges of normal and reverse phases.

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTME_m, TAUDnTRO.TAUDnTRO_m, and TAUDnTDL.TAUDnTDL_m can be changed.

Section 24 Timer Array Unit J (TAUJ)

The Timer Array Unit J (TAUJ) is multifunction timer unit with several 32-bit timers (32-bit timer array unit). TAUJ is implemented 3 units with the channel interface of 4 per one unit.

24.1 Features of RH850/P1M-E TAUJ

24.1.1 Units and Channels

This microcontroller has the following number of TAUJ units.

Table 24.1 Number of Units

| Product | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 3 | 3 |
| Name | TAUJn (n = 0 to 2) | TAUJn (n = 0 to 2) |

TAUJn has the following number of channels of timers.

Table 24.2 TAUJn Unit Configurations and Channels

| Unit Name (Channel Name) TAUJn | Channels per Unit | RH850/P1M-E 100 pins (12 ch) | RH850/P1M-E 144 pins (12 ch) |
|--------------------------------------|-------------------|------------------------------------|------------------------------------|
| TAUJ0 | 4 | √ | √ |
| TAUJ1 | 4 | √ | √ |
| TAUJ2 | 4 | √ | √ |

Table 24.3 Index

| Index | Description |
|-------|---|
| n | Throughout this section, the individual TAUJ units are identified by the index "n" (n = 0 to 2); for example, TAUJnTOM is the TAUJn channel output mode register. |
| m | The TAUJ has 4 channels. Throughout this section, the individual channels are identified by the index "m" (m = 0 to 3), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2) are denoted as CHm_even. The odd numbered channels (m = 1, 3) are denoted as CHm_odd. |

24.1.2 Register Base Address

TAUJn base addresses are listed in the following table.

TAUJn register addresses are given as offsets from the base addresses throughout the section.

Table 24.4 Register Base Addresses

| Base Address Name | Base Address |
|-------------------|------------------------|
| <TAUJ0_base> | FFE5 0000 _H |
| <TAUJ1_base> | FFE5 1000 _H |
| <TAUJ2_base> | FFE5 2000 _H |

24.1.3 Clock Supply

Clock supply by and to TAUJn is listed in the following table.

Table 24.5 Clock Supply

| Unit Name | Clock Unit Name | Supply Clock Name |
|-----------|-----------------|-------------------------------------|
| TAUJn | PCLK | High-speed peripheral clock CLK_HSB |

24.1.4 Interrupt Requests

TAUJn interrupt requests are listed in the following table.

Table 24.6 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|---------------------|------------------|------------------------|
| TAUJ0 | | | |
| INTTAUJ010 | Channel 0 interrupt | 133 | 1 |
| INTTAUJ011 | Channel 1 interrupt | 134 | 2 |
| INTTAUJ012 | Channel 2 interrupt | 135 | 3 |
| INTTAUJ013 | Channel 3 interrupt | 136 | 4 |
| TAUJ1 | | | |
| INTTAUJ110 | Channel 0 interrupt | 137 | 5 |
| INTTAUJ111 | Channel 1 interrupt | 138 | 6 |
| INTTAUJ112 | Channel 2 interrupt | 139 | 7 |
| INTTAUJ113 | Channel 3 interrupt | 140 | 8 |
| TAUJ2 | | | |
| INTTAUJ210 | Channel 0 interrupt | 256 | 9 |
| INTTAUJ211 | Channel 1 interrupt | 257 | 10 |
| INTTAUJ212 | Channel 2 interrupt | 258 | 11 |
| INTTAUJ213 | Channel 3 interrupt | 259 | 12 |

24.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

24.1.6 External Input/Output Signals

External input/output signals of TAUJn are listed below.

Table 24.7 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|--|-----|----------------------------|--|
| TAUJ0 | | | |
| TAUJ0TTIN0 to TAUJ0TTIN3 ^{*2} | I | Inputs for channel 0 to 3 | TAUJ0I0 to TAUJ0I3 |
| TAUJ0TTOUT0 to TAUJ0TTOUT3 | O | Outputs for channel 0 to 3 | TAUJ0O0 to TAUJ0O3 |
| TAUJ1 | | | |
| TAUJ1TTIN0 to TAUJ1TTIN3 ^{*2} | I | Inputs for channel 0 to 3 | TAUJ1I0 to TAUJ1I3 |
| TAUJ1TTOUT0 to TAUJ1TTOUT3 | O | Outputs for channel 0 to 3 | TAUJ1O0 to TAUJ1O3 |
| TAUJ2 | | | |
| TAUJ2TTIN0 to TAUJ2TTIN3 ^{*1*2} | I | Inputs for channel 0 to 3 | TAUJ0I0 to TAUJ0I3 TAUJ1I0 to TAUJ1I3 TAUJ2I0 to TAUJ2I3 |
| TAUJ2TTOUT0 to TAUJ2TTOUT3 | O | Outputs for channel 0 to 3 | TAUJ2O0 to TAUJ2O3 |

Note 1. The input signal is switched by PIC. For details, see **Section 29.2.2.28, PIMONSEL — Port Input Monitor Select Register** and **Section 29.2.3.14, Timer Input Monitor Function**.

Note 2. Setting of the noise filter for the port is required when the channel input pin is used. For details, see **Section 2.6, Noise Filter and Edge Level Detection Circuit**.

24.1.7 Internal Input/Output Signals

The internal input/output signals of TAUJn are listed below.

Table 24.8 Internal Input/Output Signals

| Unit Signal Name | Description | Connected to |
|--------------------------|--|--------------|
| TAUJnTSSTm ^{*1} | Simultaneous channel start trigger input | PIC |

Note 1. n = 0, 1 only. TAUJ2TSSTm is not connected to PIC.

24.2 Overview

24.2.1 Functional Overview

The TAUJ has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The timer array unit J is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the count start value or compare value.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in two operating modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels.

The synchronous operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

24.2.2 Terms

In this section, the following terms are used.

Independent channel operation function / synchronous operation channel operation function

TAUJ has 4 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented using a combination of channels.

- The independent channel operation function can be used any channel independently of all other channels.
- The synchronous channel operation function is implemented using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

Channel group

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Upper/lower channel

Based on the channel number m , a channel with a smaller channel number or higher channel number can be referred to as “upper” or “lower” channel:

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

For instance, as to channel 2, channel 1 is an upper channel and channel 3 is a lower channel. Channel 0 is the highest channel and channel 3 is the lowest channel.

24.2.3 Timer Operation Functions

This timer provides the following functions by operating individual channels independently or a combination of channels.

Table 24.9 TAUJ Operation Functions

| Operation Function | Example |
|--|------------------------|
| Independent Channel Operation Functions | Section 24.12 |
| Interval timer function | Section 24.12.1 |
| TAUJnTTINm input interval timer function | Section 24.12.2 |
| TAUJnTTINm input pulse interval measurement function | Section 24.12.3 |
| TAUJnTTINm input signal width measurement function | Section 24.12.4 |
| TAUJnTTINm input position detection function | Section 24.12.5 |
| TAUJnTTINm input period count detection function | Section 24.12.6 |
| Synchronous Channel Operation Functions | Section 24.13 |
| PWM output function | Section 24.13.1 |

24.2.4 TAUJ I/O and Interrupt Request Signals

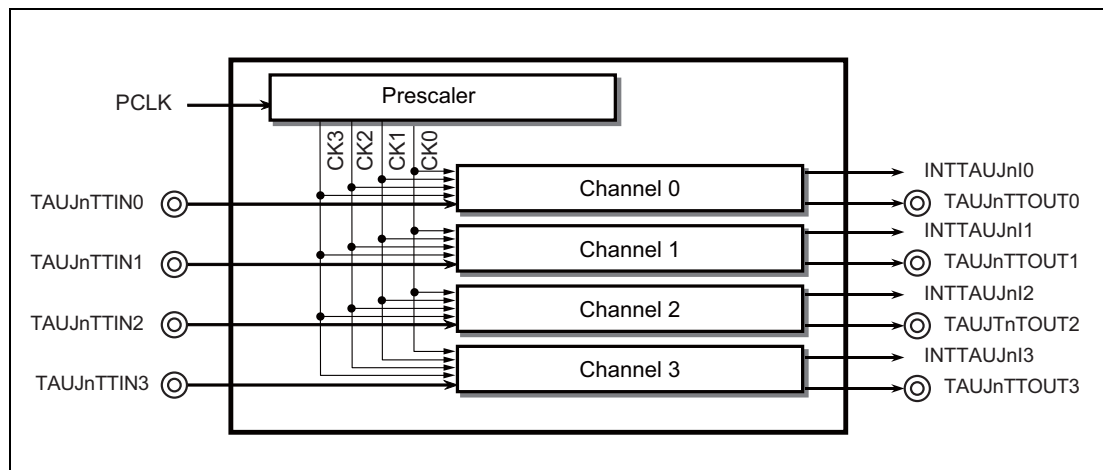


Figure 24.1 TAUJ I/O and Interrupt Request Signals

24.2.5 Block Diagram

The following figure shows the main components of the TAUJ:

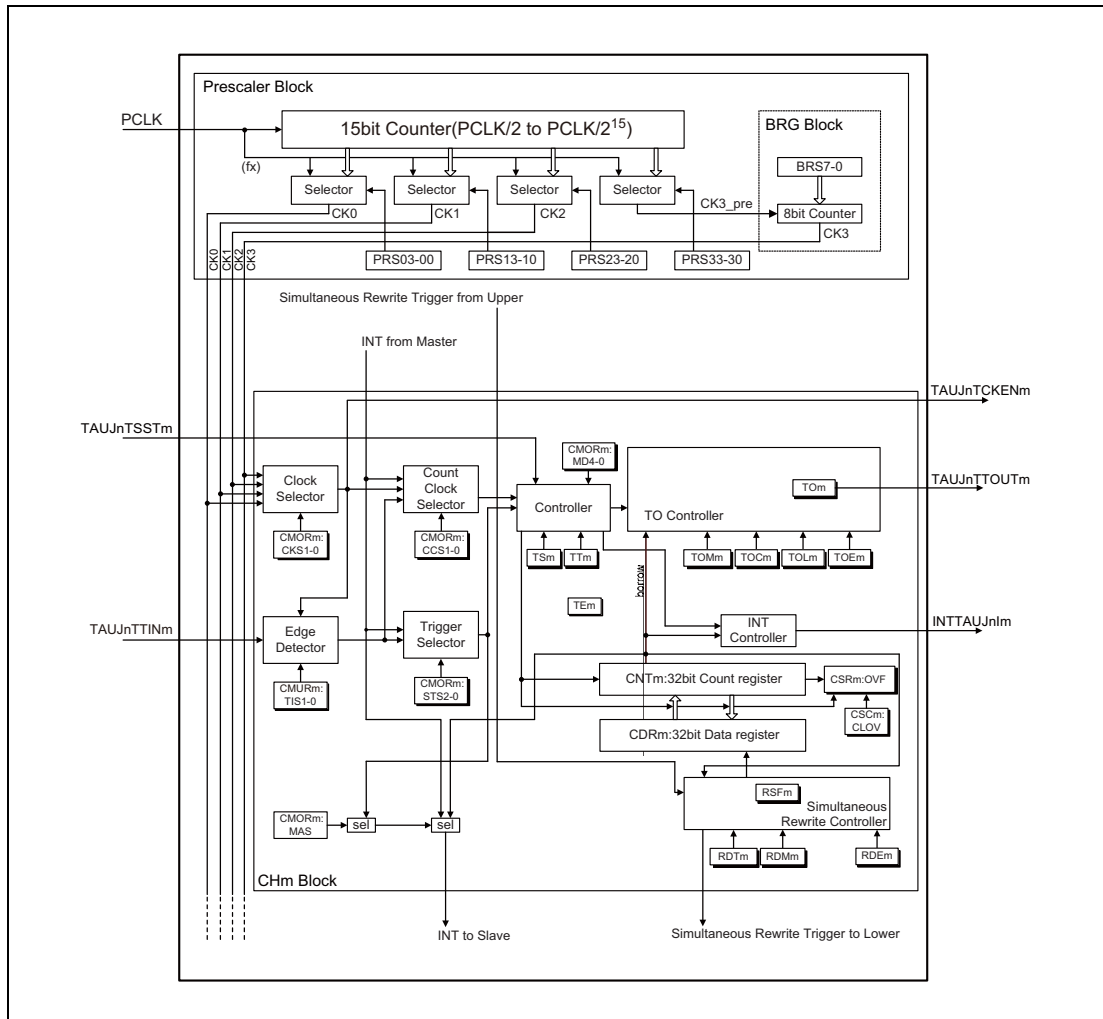


Figure 24.2 Block Diagram of the TAUJ

The prefix “TAUJn” has been omitted from the register names for the sake of clarity in the above figure.

24.2.6 Description of Blocks

The description of blocks is the followings;

Prescaler

The prescaler provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Prescaler outputs CK0 to CK2 are derived by dividing PCLK in the prescaler by a division factors of 2^0 to 2^{15} . The fourth count clock, CK3, is derived by dividing PCLK by a division factor that is not a power of 2 by using the baud rate generator.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source.

- One of the clock outputs CK0 to CK3 (selected by the clock selector)

Controller

The controller controls the main operations of the counter.

- Operating mode
(selected by the TAUJnCMORm.TAUJnMD[4:0] bits)
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

Trigger selector

The counter starts automatically when it is enabled (TAUJnTE.TAUJnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUJnTSSTm
- Valid edge of TAUJnTTINm input
- INTTAUJnIm from the master channel

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be enabled in synchronous operating modes. The data registers of all channels in a channel group (TAUJnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUJnTO controller

The output control of every channel enables the generation of various output signals such as PWM signals.

24.3 Registers

24.3.1 List of Registers

TAUJ registers are listed in the following table.

For details about <TAUJn_base>, see **Section 24.1.2, Register Base Address**.

Table 24.10 Registers

| Module | Register | Symbol | Address |
|------------------------------------|---|------------|---|
| TAUJn prescaler registers | | | |
| TAUJn | TAUJn prescaler clock select register | TAUJnTPS | <TAUJn_base> + 90 _H |
| TAUJn | TAUJn prescaler baud rate setting register | TAUJnBRS | <TAUJn_base> + 94 _H |
| TAUJn control registers | | | |
| TAUJn | TAUJn channel data register m | TAUJnCDRm | <TAUJn_base> + m × 4 _H |
| TAUJn | TAUJn channel counter register m | TAUJnCNTm | <TAUJn_base> + 10 _H + m × 4 _H |
| TAUJn | TAUJn channel mode OS register m | TAUJnCMORm | <TAUJn_base> + 80 _H + m × 4 _H |
| TAUJn | TAUJn channel mode user register m | TAUJnCMURm | <TAUJn_base> + 20 _H + m × 4 _H |
| TAUJn | TAUJn channel status register m | TAUJnCSRm | <TAUJn_base> + 30 _H + m × 4 _H |
| TAUJn | TAUJn channel status clear register m | TAUJnCSCm | <TAUJn_base> + 40 _H + m × 4 _H |
| TAUJn | TAUJn channel start trigger register | TAUJnTS | <TAUJn_base> + 54 _H |
| TAUJn | TAUJn channel enable status register | TAUJnTE | <TAUJn_base> + 50 _H |
| TAUJn | TAUJn channel stop trigger register | TAUJnTT | <TAUJn_base> + 58 _H |
| TAUJn output registers | | | |
| TAUJn | TAUJn channel output enable register | TAUJnTOE | <TAUJn_base> + 60 _H |
| TAUJn | TAUJn channel output register | TAUJnTO | <TAUJn_base> + 5C _H |
| TAUJn | TAUJn channel output mode register | TAUJnTOM | <TAUJn_base> + 98 _H |
| TAUJn | TAUJn channel output configuration register | TAUJnTOC | <TAUJn_base> + 9C _H |
| TAUJn | TAUJn channel output active level register | TAUJnTOL | <TAUJn_base> + 64 _H |
| TAUJn reload data registers | | | |
| TAUJn | TAUJn channel reload data enable register | TAUJnRDE | <TAUJn_base> + A0 _H |
| TAUJn | TAUJn channel reload data mode register | TAUJnRDM | <TAUJn_base> + A4 _H |
| TAUJn | TAUJn channel reload data trigger register | TAUJnRDT | <TAUJn_base> + 68 _H |
| TAUJn | TAUJn channel reload status register | TAUJnRSF | <TAUJn_base> + 6C _H |

24.3.2 Details of TAUJn Prescaler Registers

24.3.2.1 TAUJnTPS — TAUJn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescalers. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJnBRS.

Access: Readable/writable in 16-bit units.

Address: <TAUJn_base> + 90_H

Value after reset: FFFF_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|----------------|-----|-----|-----|----------------|-----|-----|-----|----------------|-----|-----|-----|----------------|-----|-----|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnPRS3[3:0] | | | | TAUJnPRS2[3:0] | | | | TAUJnPRS1[3:0] | | | | TAUJnPRS0[3:0] | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 24.11 TAUJnTPS Register Contents (1/3)

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 15 to 12 | TAUJnPRS3 [3:0] | Specify a CK3_PRE clock. The CK3_PRE clock is the input clock signal for the BRG unit, which supplies prescaler output CK3 to all channels. |
| | TAUJnPRS3[3:0] | CK3_PRE clock |
| | 0000 _B | PCLK/2 ⁰ |
| | 0001 _B | PCLK/2 ¹ |
| | 0010 _B | PCLK/2 ² |
| | 0011 _B | PCLK/2 ³ |
| | 0100 _B | PCLK/2 ⁴ |
| | 0101 _B | PCLK/2 ⁵ |
| | 0110 _B | PCLK/2 ⁶ |
| | 0111 _B | PCLK/2 ⁷ |
| | 1000 _B | PCLK/2 ⁸ |
| | 1001 _B | PCLK/2 ⁹ |
| | 1010 _B | PCLK/2 ¹⁰ |
| | 1011 _B | PCLK/2 ¹¹ |
| | 1100 _B | PCLK/2 ¹² |
| | 1101 _B | PCLK/2 ¹³ |
| | 1110 _B | PCLK/2 ¹⁴ |
| | 1111 _B | PCLK/2 ¹⁵ |

The above bits are rewritable only when all the counters using CK3 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 24.11 TAUJnTPS Register Contents (2/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|----------------------|---|----------------|-----------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|
| 11 to 8 | TAUJnPRS2 [3:0] | Specify the output CK2 clock. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJnPRS2[3:0]</th> <th>CK2 Clock</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>PCLK/2⁰</td></tr> <tr><td>0001_B</td><td>PCLK/2¹</td></tr> <tr><td>0010_B</td><td>PCLK/2²</td></tr> <tr><td>0011_B</td><td>PCLK/2³</td></tr> <tr><td>0100_B</td><td>PCLK/2⁴</td></tr> <tr><td>0101_B</td><td>PCLK/2⁵</td></tr> <tr><td>0110_B</td><td>PCLK/2⁶</td></tr> <tr><td>0111_B</td><td>PCLK/2⁷</td></tr> <tr><td>1000_B</td><td>PCLK/2⁸</td></tr> <tr><td>1001_B</td><td>PCLK/2⁹</td></tr> <tr><td>1010_B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011_B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100_B</td><td>PCLK/2¹²</td></tr> <tr><td>1101_B</td><td>PCLK/2¹³</td></tr> <tr><td>1110_B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111_B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> | TAUJnPRS2[3:0] | CK2 Clock | 0000 _B | PCLK/2 ⁰ | 0001 _B | PCLK/2 ¹ | 0010 _B | PCLK/2 ² | 0011 _B | PCLK/2 ³ | 0100 _B | PCLK/2 ⁴ | 0101 _B | PCLK/2 ⁵ | 0110 _B | PCLK/2 ⁶ | 0111 _B | PCLK/2 ⁷ | 1000 _B | PCLK/2 ⁸ | 1001 _B | PCLK/2 ⁹ | 1010 _B | PCLK/2 ¹⁰ | 1011 _B | PCLK/2 ¹¹ | 1100 _B | PCLK/2 ¹² | 1101 _B | PCLK/2 ¹³ | 1110 _B | PCLK/2 ¹⁴ | 1111 _B | PCLK/2 ¹⁵ |
| TAUJnPRS2[3:0] | CK2 Clock | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | PCLK/2 ⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | PCLK/2 ¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | PCLK/2 ² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 _B | PCLK/2 ³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 _B | PCLK/2 ⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 _B | PCLK/2 ⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 _B | PCLK/2 ⁶ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 _B | PCLK/2 ⁷ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 _B | PCLK/2 ⁸ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 _B | PCLK/2 ⁹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 _B | PCLK/2 ¹⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 _B | PCLK/2 ¹¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 _B | PCLK/2 ¹² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 _B | PCLK/2 ¹³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 _B | PCLK/2 ¹⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | PCLK/2 ¹⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| The above bits are rewritable only when all the counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0). | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 to 4 | TAUJnPRS1 [3:0] | Specify the output CK1 clock. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJnPRS1[3:0]</th> <th>CK1 Clock</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>PCLK/2⁰</td></tr> <tr><td>0001_B</td><td>PCLK/2¹</td></tr> <tr><td>0010_B</td><td>PCLK/2²</td></tr> <tr><td>0011_B</td><td>PCLK/2³</td></tr> <tr><td>0100_B</td><td>PCLK/2⁴</td></tr> <tr><td>0101_B</td><td>PCLK/2⁵</td></tr> <tr><td>0110_B</td><td>PCLK/2⁶</td></tr> <tr><td>0111_B</td><td>PCLK/2⁷</td></tr> <tr><td>1000_B</td><td>PCLK/2⁸</td></tr> <tr><td>1001_B</td><td>PCLK/2⁹</td></tr> <tr><td>1010_B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011_B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100_B</td><td>PCLK/2¹²</td></tr> <tr><td>1101_B</td><td>PCLK/2¹³</td></tr> <tr><td>1110_B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111_B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> | TAUJnPRS1[3:0] | CK1 Clock | 0000 _B | PCLK/2 ⁰ | 0001 _B | PCLK/2 ¹ | 0010 _B | PCLK/2 ² | 0011 _B | PCLK/2 ³ | 0100 _B | PCLK/2 ⁴ | 0101 _B | PCLK/2 ⁵ | 0110 _B | PCLK/2 ⁶ | 0111 _B | PCLK/2 ⁷ | 1000 _B | PCLK/2 ⁸ | 1001 _B | PCLK/2 ⁹ | 1010 _B | PCLK/2 ¹⁰ | 1011 _B | PCLK/2 ¹¹ | 1100 _B | PCLK/2 ¹² | 1101 _B | PCLK/2 ¹³ | 1110 _B | PCLK/2 ¹⁴ | 1111 _B | PCLK/2 ¹⁵ |
| TAUJnPRS1[3:0] | CK1 Clock | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | PCLK/2 ⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | PCLK/2 ¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | PCLK/2 ² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 _B | PCLK/2 ³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 _B | PCLK/2 ⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 _B | PCLK/2 ⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 _B | PCLK/2 ⁶ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 _B | PCLK/2 ⁷ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 _B | PCLK/2 ⁸ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 _B | PCLK/2 ⁹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 _B | PCLK/2 ¹⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 _B | PCLK/2 ¹¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 _B | PCLK/2 ¹² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 _B | PCLK/2 ¹³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 _B | PCLK/2 ¹⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | PCLK/2 ¹⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| The above bits are rewritable only when all the counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0). | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 24.11 TAUJnTPS Register Contents (3/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|----------------------|---|----------------|-----------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|-------------------|----------------------|
| 3 to 0 | TAUJnPRS0 [3:0] | Specify a CK0 clock. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJnPRS0[3:0]</th> <th>CK0 Clock</th> </tr> </thead> <tbody> <tr><td>0000_B</td><td>PCLK/2⁰</td></tr> <tr><td>0001_B</td><td>PCLK/2¹</td></tr> <tr><td>0010_B</td><td>PCLK/2²</td></tr> <tr><td>0011_B</td><td>PCLK/2³</td></tr> <tr><td>0100_B</td><td>PCLK/2⁴</td></tr> <tr><td>0101_B</td><td>PCLK/2⁵</td></tr> <tr><td>0110_B</td><td>PCLK/2⁶</td></tr> <tr><td>0111_B</td><td>PCLK/2⁷</td></tr> <tr><td>1000_B</td><td>PCLK/2⁸</td></tr> <tr><td>1001_B</td><td>PCLK/2⁹</td></tr> <tr><td>1010_B</td><td>PCLK/2¹⁰</td></tr> <tr><td>1011_B</td><td>PCLK/2¹¹</td></tr> <tr><td>1100_B</td><td>PCLK/2¹²</td></tr> <tr><td>1101_B</td><td>PCLK/2¹³</td></tr> <tr><td>1110_B</td><td>PCLK/2¹⁴</td></tr> <tr><td>1111_B</td><td>PCLK/2¹⁵</td></tr> </tbody> </table> | TAUJnPRS0[3:0] | CK0 Clock | 0000 _B | PCLK/2 ⁰ | 0001 _B | PCLK/2 ¹ | 0010 _B | PCLK/2 ² | 0011 _B | PCLK/2 ³ | 0100 _B | PCLK/2 ⁴ | 0101 _B | PCLK/2 ⁵ | 0110 _B | PCLK/2 ⁶ | 0111 _B | PCLK/2 ⁷ | 1000 _B | PCLK/2 ⁸ | 1001 _B | PCLK/2 ⁹ | 1010 _B | PCLK/2 ¹⁰ | 1011 _B | PCLK/2 ¹¹ | 1100 _B | PCLK/2 ¹² | 1101 _B | PCLK/2 ¹³ | 1110 _B | PCLK/2 ¹⁴ | 1111 _B | PCLK/2 ¹⁵ |
| TAUJnPRS0[3:0] | CK0 Clock | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0000 _B | PCLK/2 ⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0001 _B | PCLK/2 ¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0010 _B | PCLK/2 ² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0011 _B | PCLK/2 ³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0100 _B | PCLK/2 ⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 _B | PCLK/2 ⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0110 _B | PCLK/2 ⁶ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0111 _B | PCLK/2 ⁷ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000 _B | PCLK/2 ⁸ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1001 _B | PCLK/2 ⁹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1010 _B | PCLK/2 ¹⁰ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1011 _B | PCLK/2 ¹¹ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1100 _B | PCLK/2 ¹² | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1101 _B | PCLK/2 ¹³ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1110 _B | PCLK/2 ¹⁴ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1111 _B | PCLK/2 ¹⁵ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The above bits are rewritable only when all the counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).

NOTE

TAUJn clock input PCLK is defined in the first part of this section, **Section 24.1.3, Clock Supply**.

24.3.2.2 TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUJnTPS. TAUJnPRS3[3:0].

Access: Readable/writable in 8-bit units.

Address: <TAUJn_base> + 94_H

Value after reset: 00_H This register is initialized by various types of reset.

| | | | | | | | | |
|-------------------|---------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnBRS[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 24.12 TAUJnBRS Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | |
|------------------------|----------------|---|---------------|-----------|------------------------|-------------|------------------------|-------------|------------------------|-------------|------------------------|-------------|-----|-----|------------------------|---------------|------------------------|---------------|
| 7 to 0 | TAUJnBRS [7:0] | Specify a CK3_PRE clock division factor for generating CK3. | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJnBRS[7:0]</th> <th>CK3 clock</th> </tr> </thead> <tbody> <tr> <td>0000 0000_B</td> <td>CK3_PRE / 1</td> </tr> <tr> <td>0000 0001_B</td> <td>CK3_PRE / 2</td> </tr> <tr> <td>0000 0010_B</td> <td>CK3_PRE / 3</td> </tr> <tr> <td>0000 0011_B</td> <td>CK3_PRE / 4</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>1111 1110_B</td> <td>CK3_PRE / 255</td> </tr> <tr> <td>1111 1111_B</td> <td>CK3_PRE / 256</td> </tr> </tbody> </table> | TAUJnBRS[7:0] | CK3 clock | 0000 0000 _B | CK3_PRE / 1 | 0000 0001 _B | CK3_PRE / 2 | 0000 0010 _B | CK3_PRE / 3 | 0000 0011 _B | CK3_PRE / 4 | ... | ... | 1111 1110 _B | CK3_PRE / 255 | 1111 1111 _B | CK3_PRE / 256 |
| TAUJnBRS[7:0] | CK3 clock | | | | | | | | | | | | | | | | | |
| 0000 0000 _B | CK3_PRE / 1 | | | | | | | | | | | | | | | | | |
| 0000 0001 _B | CK3_PRE / 2 | | | | | | | | | | | | | | | | | |
| 0000 0010 _B | CK3_PRE / 3 | | | | | | | | | | | | | | | | | |
| 0000 0011 _B | CK3_PRE / 4 | | | | | | | | | | | | | | | | | |
| ... | ... | | | | | | | | | | | | | | | | | |
| 1111 1110 _B | CK3_PRE / 255 | | | | | | | | | | | | | | | | | |
| 1111 1111 _B | CK3_PRE / 256 | | | | | | | | | | | | | | | | | |

24.3.3 Details of TAUJn Control Registers

24.3.3.1 TAUJnCDRm — TAUJn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJnCMORm.TAUJnMD[4:1].

- Access:** Readable/writable in 32-bit units.
- When this register functions as a capture register, only reading is possible. Any write operation is ignored.
 - When this register functions as a compare register, reading and writing is possible.

Address: <TAUJn_base> + 0_H + m × 4_H

Value after reset: 0000 0000_H This register is initialized by various types of reset.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TAUJnCDR[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCDR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 24.13 TAUJnCDRm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|--|
| 31 to 0 | TAUJnCDR [31:0] | Data register for capture/compare values |

24.3.3.2 TAUJnCNTm — TAUJn Channel Counter Register

This is a channel m counter register.

Access: Only readable in 32-bit units.

Address: <TAUJn_base> + 10_H + m × 4_H

Value after reset: FFFF FFFF_H This register is initialized by various types of reset.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| TAUJnCNT[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TAUJnCNT[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 24.14 TAUJnCNTm Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|----------------------|
| 31 to 0 | TAUJnCNT [31:0] | 32-bit counter value |

The read value depends on a counter, an operating mode change, or TAUJnTS.TAUJnTSm/TAUJnTT.TAUJnTTm bit value.

The initial counter read value depends on the operating mode and how the counter is stopped.

- By a reset
- By a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUJnTE.TAUJnTEm = 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) with the counter waiting for a start trigger.

Table 24.15 TAUJnCNTm Read Values after Re-Enabling Counter

| Mode Name | Count Method (Up/Down) | TAUJnCNTm | | |
|-----------------------------|------------------------|---------------------------|--------------------|-------------------------------|
| | | Start Value ^{*1} | After Stop Trigger | After One Count |
| Interval timer mode | Count down | FFFF FFFF _H | Stop value | — |
| Capture mode | Count up | 0000 0000 _H | Stop value | — |
| One-count mode | Count down | FFFF FFFF _H | Stop value | Stop value |
| Capture and one-count mode | Count up | 0000 0000 _H | Stop value | Capture value + 1 (TAUJnCDRm) |
| Count capture mode | Count up | 0000 0000 _H | Stop value | — |
| Capture and gate count mode | Count up | 0000 0000 _H | Stop value | Stop value |

Note 1. The value set in the TAUJnCNTm when operation modes are changed after reset release.

24.3.3.3 TAUJnCMORm — TAUJn Channel Mode OS Register

This register controls channel m operation.

Access: Readable/writable in 16-bit units. Writable only when the counter is stopped (TAUJnTE.TAUJnTE_m = 0).

Address: <TAUJn_base> + 80_H + m × 4_H

Value after reset: 0000_H This register is initialized by various types of reset.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|-----|
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | | TAUJnCOS[1:0] | | — | TAUJnMD[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.16 TAUJnCMORm Register Contents (1/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|---------------|---|-----------|-----------|------------------------------|---|---|---|---|---|--------------------|---|---|-----|---|---|-----|
| 15, 14 | TAUJnCKS[1:0] | <p>Selects the operating clock for use with the TAUJnTTIN_m input edge detection circuit.</p> <p>According to the setting of the TAUJnCMOR_m.TAUJnCCS[1:0] bits, this can also be used as the clock to drive counting by TAUJnCNT_m.</p> <table border="1"> <thead> <tr> <th>TAUJnCKS1</th> <th>TAUJnCKS0</th> <th>Selection of Operating Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>CK0</td> </tr> <tr> <td>0</td> <td>1</td> <td>CK1</td> </tr> <tr> <td>1</td> <td>0</td> <td>CK2</td> </tr> <tr> <td>1</td> <td>1</td> <td>CK3</td> </tr> </tbody> </table> | TAUJnCKS1 | TAUJnCKS0 | Selection of Operating Clock | 0 | 0 | CK0 | 0 | 1 | CK1 | 1 | 0 | CK2 | 1 | 1 | CK3 |
| TAUJnCKS1 | TAUJnCKS0 | Selection of Operating Clock | | | | | | | | | | | | | | | |
| 0 | 0 | CK0 | | | | | | | | | | | | | | | |
| 0 | 1 | CK1 | | | | | | | | | | | | | | | |
| 1 | 0 | CK2 | | | | | | | | | | | | | | | |
| 1 | 1 | CK3 | | | | | | | | | | | | | | | |
| 13, 12 | TAUJnCCS[1:0] | <p>Selects a count clock for TAUJnCNT_m counter.</p> <table border="1"> <thead> <tr> <th>TAUJnCCS1</th> <th>TAUJnCCS0</th> <th>Selection of Operating Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Operating clock specified by TAUJnCMOR_m.TAUJnCKS[1:0].</td> </tr> <tr> <td>0</td> <td>1</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table> | TAUJnCCS1 | TAUJnCCS0 | Selection of Operating Clock | 0 | 0 | Operating clock specified by TAUJnCMOR _m .TAUJnCKS[1:0]. | 0 | 1 | Setting prohibited | 1 | 0 | | 1 | 1 | |
| TAUJnCCS1 | TAUJnCCS0 | Selection of Operating Clock | | | | | | | | | | | | | | | |
| 0 | 0 | Operating clock specified by TAUJnCMOR _m .TAUJnCKS[1:0]. | | | | | | | | | | | | | | | |
| 0 | 1 | Setting prohibited | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | | | | | | |
| 11 | TAUJnMAS | <p>Specifies whether the channel is a master or slave channel during synchronous channel operation.</p> <p>0: Slave 1: Master</p> <p>This bit setting is valid only for even channels (CH_m_even). Odd channels (CH_m-odd) are fixed to 0.</p> | | | | | | | | | | | | | | | |

Table 24.16 TAUJnCMORm Register Contents (2/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---------------|---|---|------------|-----------|------------------------|---|---|--|---|---|---|---|--|---|---|---|--|---|---|---|---|---|---|---|--|---|---|---|--------------------|---|---|---|--|---|---|---|--|
| 10 to 8 | TAUJnSTS[2:0] | Selects an external start trigger. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJnSTS2</th> <th>TAUJnS TS1</th> <th>TAUJnSTS0</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Software trigger</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Valid edge of TAUJnTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Valid edge of TAUJnTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>INTTAUJnIm of master channel is used as a start trigger.</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table> | TAUJnSTS2 | TAUJnS TS1 | TAUJnSTS0 | Functional Description | 0 | 0 | 0 | Software trigger | 0 | 0 | 1 | Valid edge of TAUJnTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0]. | 0 | 1 | 0 | Valid edge of TAUJnTTINm input signal is used as a start trigger and the reverse edge as a stop trigger. | 0 | 1 | 1 | Setting prohibited | 1 | 0 | 0 | INTTAUJnIm of master channel is used as a start trigger. | 1 | 0 | 1 | Setting prohibited | 1 | 1 | 0 | | 1 | 1 | 1 | |
| TAUJnSTS2 | TAUJnS TS1 | TAUJnSTS0 | Functional Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | Software trigger | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | Valid edge of TAUJnTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0]. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | Valid edge of TAUJnTTINm input signal is used as a start trigger and the reverse edge as a stop trigger. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | INTTAUJnIm of master channel is used as a start trigger. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7, 6 | TAUJnCOS[1:0] | Specify the timing for updating capture register TAUJnCDRm and overflow flag TAUJnCSRm. TAUJnOVF of channel m. These bits are valid only when channel m is in capture function (capture mode or capture & one-count mode). | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJnCOS1</th> <th>TAUJnCOS0</th> <th>TAUJnCDRm</th> <th>TAUJnCSRm.TAUJnOVF</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Updated when valid edge of TAUJnTTINm input is detected.</td> <td>Updated (cleared or set) when valid edge of TAUJnTTINm input is detected: <ul style="list-style-type: none"> Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected. Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected. </td> </tr> <tr> <td>0</td> <td>1</td> <td></td> <td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Updated when valid edge of TAUJnTTINm input is detected and when a counter overflow occurs.</td> <td>No setting</td> </tr> <tr> <td>1</td> <td>1</td> <td> <ul style="list-style-type: none"> Detection of valid edge of TAUJnTTINm input: The counter value is written into TAUJnCDRm. Occurrence of overflow: FFFF FFFF_H is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJnTTINm is ignored. </td> <td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td> </tr> </tbody> </table> | TAUJnCOS1 | TAUJnCOS0 | TAUJnCDRm | TAUJnCSRm.TAUJnOVF | 0 | 0 | Updated when valid edge of TAUJnTTINm input is detected. | Updated (cleared or set) when valid edge of TAUJnTTINm input is detected: <ul style="list-style-type: none"> Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected. Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected. | 0 | 1 | | Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1. | 1 | 0 | Updated when valid edge of TAUJnTTINm input is detected and when a counter overflow occurs. | No setting | 1 | 1 | <ul style="list-style-type: none"> Detection of valid edge of TAUJnTTINm input: The counter value is written into TAUJnCDRm. Occurrence of overflow: FFFF FFFF_H is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJnTTINm is ignored. | Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1. | | | | | | | | | | | | | | | | |
| TAUJnCOS1 | TAUJnCOS0 | TAUJnCDRm | TAUJnCSRm.TAUJnOVF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Updated when valid edge of TAUJnTTINm input is detected. | Updated (cleared or set) when valid edge of TAUJnTTINm input is detected: <ul style="list-style-type: none"> Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected. Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | | Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Updated when valid edge of TAUJnTTINm input is detected and when a counter overflow occurs. | No setting | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | <ul style="list-style-type: none"> Detection of valid edge of TAUJnTTINm input: The counter value is written into TAUJnCDRm. Occurrence of overflow: FFFF FFFF_H is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJnTTINm is ignored. | Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 24.16 TAUJnCMORm Register Contents (3/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|---|-----------|----------------------|---|--|----------------|--|---|---|---|---|-----|---------------------|---|---|---|---|-----|--------------------|---|---|---|---|-----|--------------|---|---|---|---|-----|--------------------|---|---|---|---|-----|----------------|---|---|---|---|-----|--------------------|---|---|---|---|-----|----------------------------|---|---|---|---|-----|--------------------|---|---|---|---|--|--|---|---|---|---|--|--|---|---|---|---|--|--|---|---|---|---|-----|--------------------|---|---|---|---|---|--------------------|---|---|---|---|---|-----------------------------|------------------|--|--|--|--|--------------------|
| 4 to 0 | TAUJnMD[4:0] | Specify an operating mode. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJn MD4</th> <th>TAUJn MD3</th> <th>TAUJn MD2</th> <th>TAUJn MD1</th> <th>TAUJn MD0</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1/0</td> <td>Interval timer mode</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1/0</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1/0</td> <td>Capture mode</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1/0</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1/0</td> <td>One-count mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1/0</td> <td>Setting prohibited</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1/0</td> <td>Capture and one-count mode</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1/0</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>1/0</td> <td>Count capture mode</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>Capture and gate count mode</td> </tr> <tr> <td colspan="5">Other than above</td> <td>Setting prohibited</td> </tr> </tbody> </table> | TAUJn MD4 | TAUJn MD3 | TAUJn MD2 | TAUJn MD1 | TAUJn MD0 | Functional Description | 0 | 0 | 0 | 0 | 1/0 | Interval timer mode | 0 | 0 | 0 | 1 | 1/0 | Setting prohibited | 0 | 0 | 1 | 0 | 1/0 | Capture mode | 0 | 0 | 1 | 1 | 1/0 | Setting prohibited | 0 | 1 | 0 | 0 | 1/0 | One-count mode | 0 | 1 | 0 | 1 | 1/0 | Setting prohibited | 0 | 1 | 1 | 0 | 1/0 | Capture and one-count mode | 0 | 1 | 1 | 1 | 1/0 | Setting prohibited | 1 | 0 | 0 | 0 | | | 1 | 0 | 0 | 1 | | | 1 | 0 | 1 | 0 | | | 1 | 0 | 1 | 1 | 1/0 | Count capture mode | 1 | 1 | 0 | 0 | 0 | Setting prohibited | 1 | 1 | 0 | 1 | 0 | Capture and gate count mode | Other than above | | | | | Setting prohibited |
| TAUJn MD4 | TAUJn MD3 | TAUJn MD2 | TAUJn MD1 | TAUJn MD0 | Functional Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 1/0 | Interval timer mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1/0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 1/0 | Capture mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 1/0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 1/0 | One-count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 1/0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 1/0 | Capture and one-count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 1/0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | 1/0 | Count capture mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 1 | 0 | Capture and gate count mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Other than above | | | | | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>Mode</th> <th>Role of TAUJnMD0 Bit</th> </tr> </thead> <tbody> <tr> <td>Interval timer mode Capture mode Count capture mode</td> <td>Specifies whether INTTAUJnIm is output at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnIm is not output. 1: INTTAUJnIm is output.</td> </tr> <tr> <td>One-count mode</td> <td>Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection.</td> </tr> <tr> <td></td> <td>CAUTION In one-count mode, the INTTAUJnIm input signal is not output when the counter starts.</td> </tr> <tr> <td>Capture and one-count mode Capture and gate count mode</td> <td>Set this bit to 0. CAUTION The INTTAUDnIm input signal is not output when the counter starts. The start trigger detected during count operation is disabled.</td> </tr> </tbody> </table> | Mode | Role of TAUJnMD0 Bit | Interval timer mode Capture mode Count capture mode | Specifies whether INTTAUJnIm is output at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnIm is not output. 1: INTTAUJnIm is output. | One-count mode | Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. | | CAUTION In one-count mode, the INTTAUJnIm input signal is not output when the counter starts. | Capture and one-count mode Capture and gate count mode | Set this bit to 0. CAUTION The INTTAUDnIm input signal is not output when the counter starts. The start trigger detected during count operation is disabled. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mode | Role of TAUJnMD0 Bit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Interval timer mode Capture mode Count capture mode | Specifies whether INTTAUJnIm is output at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnIm is not output. 1: INTTAUJnIm is output. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| One-count mode | Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | CAUTION In one-count mode, the INTTAUJnIm input signal is not output when the counter starts. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Capture and one-count mode Capture and gate count mode | Set this bit to 0. CAUTION The INTTAUDnIm input signal is not output when the counter starts. The start trigger detected during count operation is disabled. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

24.3.3.4 TAUJnCMURm — TAUJn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUJnTTINm input.

Access: Readable/writable in 8-bit units.

Address: <TAUJn_base> + 20_H + m × 4_H

Value after reset: 00_H This register is initialized by various types of reset.

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 24.17 TAUJnCMURm Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|---------------|--|-----------|-----------|------------------------|---|---|--------------|---|---|-------------|---|---|--|---|---|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 1, 0 | TAUJnTIS[1:0] | Specify a valid edge of TAUJnTTINm input signal. | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TAUJnTIS1</th> <th>TAUJnTIS0</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Falling edge</td> </tr> <tr> <td>0</td> <td>1</td> <td>Rising edge</td> </tr> <tr> <td>1</td> <td>0</td> <td>Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge</td> </tr> <tr> <td>1</td> <td>1</td> <td>Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge</td> </tr> </tbody> </table> | TAUJnTIS1 | TAUJnTIS0 | Functional Description | 0 | 0 | Falling edge | 0 | 1 | Rising edge | 1 | 0 | Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge | 1 | 1 | Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge |
| TAUJnTIS1 | TAUJnTIS0 | Functional Description | | | | | | | | | | | | | | | |
| 0 | 0 | Falling edge | | | | | | | | | | | | | | | |
| 0 | 1 | Rising edge | | | | | | | | | | | | | | | |
| 1 | 0 | Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge | | | | | | | | | | | | | | | |
| 1 | 1 | Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge | | | | | | | | | | | | | | | |

Edge detection of TAUJnTTINm input signal is based on the operating clock selected by TAUJnCMORm.TAUJnCKS[1:0].

24.3.3.5 TAUJnCSRm — TAUJn Channel Status Register

This register indicates the overflow status of channel m.

Access: Only readable in 8-bit units.

Address: <TAUJn_base> + 30_H + m × 4_H

Value after reset: 0X_H This register is initialized by various types of reset.

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TAUJnOVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | — | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 24.18 TAUJnCSRm Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. |
| 1 | Reserved | When read, an undefined value is read. |
| 0 | TAUJnOVF | Indicates the counter overflow status: 0: No overflow occurs 1: Overflow occurs This bit is used only in the following modes: <ul style="list-style-type: none"> • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOSC[1:0]. |

24.3.3.6 TAUJnCSCm — TAUJn Channel Status Clear Trigger Register

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

Access: Only writable in 8-bit units. The read value is always 00_H.

Address: <TAUJn_base> + 40_H + m × 4_H

Value after reset: 00_H This register is initialized by various types of reset.

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|-----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TAUJnCLOV |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 24.19 TAUJnCSCm Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TAUJnCLOV | 0: No effect (Setting 0 does not affect the overflow flag TAUJnCSRm.TAUJnOVF bit.) 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF |

24.3.3.7 TAUJnTS — TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

Access: Only writable in 8-bit units. The read value is always 00_H.

Address: <TAUJn_base> + 54_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|-----------|-----------|-----------|-----------|
| | — | — | — | — | TAUJnTS03 | TAUJnTS02 | TAUJnTS01 | TAUJnTS00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | W | W | W | W |

Table 24.20 TAUJnTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 4 | Reserved | When writing, write the value after reset. |
| 3 to 0 | TAUJnTSm | Enables the counter operation for channel m: 0: No effect (Setting 0 does not enable the counter operation of channel m.) 1: Enables the counter operation and sets TAUJnTE.TAUJnTEm = 1 |

Only the counter operation is enabled even if TAUJnTE.TAUJnTEm = 1.
Whether the counter is started or not depends on the selected of operating mode.

24.3.3.8 TAUJnTE — TAUJn Channel Enable Status Register

This register enables/disables a counter operation.

Access: Only readable in 8-bit units.

Address: <TAUJn_base> + 50_H

Value after reset: 00_H This register is initialized by various types of reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|-----------|-----------|-----------|-----------|
| | — | — | — | — | TAUJnTE03 | TAUJnTE02 | TAUJnTE01 | TAUJnTE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 24.21 TAUJnTE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. |
| 3 to 0 | TAUJnTEm | Enables/disables channel m's counter operation. 0: Disables the counter operation 1: Enables the counter operation |

These bits are set to 1 when trigger input of TAUJnTSSTm (synchronous channel start trigger signal) is detected or when TAUJnTS.TAUJnTSm is set to 1.
These bits are set to 0 when TAUJnTT.TAUJnTTm is set to 1.

24.3.3.9 TAUJnTT — TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: Only writable in 8-bit units. The read value is always 00_H.

Address: <TAUJn_base> + 58_H

Value after reset: 00_H This register is initialized by various types of reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|-----------|-----------|-----------|-----------|
| | — | — | — | — | TAUJnTT03 | TAUJnTT02 | TAUJnTT01 | TAUJnTT00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | W | W | W | W |

Table 24.22 TAUJnTT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 4 | Reserved | When writing, write the value after reset. |
| 3 to 0 | TAUJnTTm | Stops channel m's counter operation. 0: No effect (Setting 0 does not enable the counter operation of channel m.) 1: Stops the counter operation and resets TAUJnTE.TAUJnTEm TAUJnCNTm, TAUJnTO.TAUJnTOM, and TAUJnTTOUTm retain the values provided before the counter is stopped. |

24.3.4 TAUJn Simultaneous Rewrite Register Details

24.3.4.1 TAUJnRDE — TAUJn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUJnCDRm. It also enables and disables simultaneous rewrite of the data register TAUJnTOL.TAUJnTOLm for the PWM output function.

Access: This register can be read/written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + A0_H

Value after reset: 00_H This register is initialized by various types of reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| | — | — | — | — | TAUJnRDE03 | TAUJnRDE02 | TAUJnRDE01 | TAUJnRDE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 24.23 TAUJnRDE Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | TAUJnRDEm | Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite |

24.3.4.2 TAUJnRDM — TAUJn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

Access: This register can be read/written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + A4_H

Value after reset: 00_H This register is initialized by various types of reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| | — | — | — | — | TAUJnRDM03 | TAUJnRDM02 | TAUJnRDM01 | TAUJnRDM00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 24.24 TAUJnRDM Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | TAUJnRDMm | Specify when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: Setting prohibited |

These bits only apply when TAUJnRDE.TAUJnRDEm = 1.

24.3.4.3 TAUJnRDT — TAUJn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enable state.

Access: This register can only be written in 8-bit unit. It is always read as 00_H.

Address: <TAUJn_base> + 68_H

Value after reset: 00_H This register is initialized by any reset source.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| | — | — | — | — | TAUJnRDT03 | TAUJnRDT02 | TAUJnRDT01 | TAUJnRDT00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | W | W | W | W |

Table 24.25 TAUJnRDT Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 4 | Reserved | When writing, write the value after reset. |
| 3 to 0 | TAUJnRDTm | Triggers the simultaneous rewrite enable state: 0: No effect (Setting 0 does not trigger the simultaneous rewrite enable state.) 1: Simultaneous rewrite enable state is triggered. The simultaneous rewrite enable flag (TAUJnRSF.TAUJnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when TAUJnRDE.TAUJnRDEm = 1. |

24.3.4.4 TAUJnRSF — TAUJn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register can only be read in 8-bit units.

Address: <TAUJn_base> + 6C_H

Value after reset: 00_H This register is initialized by any reset source.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| | — | — | — | — | TAUJnRSF03 | TAUJnRSF02 | TAUJnRSF01 | TAUJnRSF00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 24.26 TAUJnRSF Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. |
| 3 to 0 | TAUJnRSFm | Indicates the simultaneous rewrite status: 0: Indicates that the simultaneous rewrite is complete by the generation of the simultaneous rewrite trigger. 1: Indicates a trigger waiting state in the simultaneous rewrite enable state (TAUJnRDT.TAUJnRDTm = 1). |

24.3.5 TAUJn Output Registers Details

24.3.5.1 TAUJnTOE — TAUJn Channel Output Enable Register

This register enables and disables Independent Channel Output Mode Controlled by Software.

Access: This register can be read/written in 8-bit units.

Address: <TAUJn_base> + 60_H

Value after reset: 00_H This register is initialized by various types of reset.

| | | | | | | | | |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | TAUJnTOE03 | TAUJnTOE02 | TAUJnTOE01 | TAUJnTOE00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 24.27 TAUJnTOE Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | TAUJnTOEm | Enables/disables Independent Channel Output Mode: 0: Disables Independent Timer Output Function (by software) 1: Enables Independent Timer Output Function |

24.3.5.2 TAUJnTO — TAUJn Channel Output Register

This register specifies and reads the level of TAUJnTTOUTm.

Access: This register can be read/written in 8-bit units.

Address: <TAUJn_base> + 5C_H

Value after reset: 00_H This register is initialized by various types of reset.

| | | | | | | | | |
|-------------------|---|---|---|---|-----------|-----------|-----------|-----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | TAUJnTO03 | TAUJnTO02 | TAUJnTO01 | TAUJnTO00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 24.28 TAUJnTO Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | TAUJnTOm | Specifies/reads the level of TAUJnTTOUTm: 0: Low 1: High Only TAUJnTO.TAUJnTOm bits for which the independent channel output function is disabled (TAUJnTOE.TAUJnTOEm = 0) can be written. |

24.3.5.3 TAUJnTOM — TAUJn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read/written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 98_H

Value after reset: 00_H This register is initialized by various types of reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| | — | — | — | — | TAUJnTOM03 | TAUJnTOM02 | TAUJnTOM01 | TAUJnTOM00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 24.29 TAUJnTOM Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | TAUJnTOMm | Specify the output mode: 0: Independent channel output mode 1: Synchronous channel output mode The output mode depends on several channel output control bit TAUJnTOE.TAUJnTOEm setting. |

24.3.5.4 TAUJnTOC — TAUJn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUJnTOM.TAUJnTOMm.

Access: This register can be read/written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 9C_H

Value after reset: 00_H This register is initialized by various types of reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| | — | — | — | — | TAUJnTOC03 | TAUJnTOC02 | TAUJnTOC01 | TAUJnTOC00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 24.30 TAUJnTOC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | TAUJnTOCm | Specify the output mode: 0: Operation mode 1 (= Toggle mode) 1: Setting prohibited |

These bits must always be set 0 in all output modes except the independent channel output mode controlled by software. The output mode depends on the setting of TAUJnTOM.TAUJnTOMm as shown in the table below.

| TAUJn TOMm | TAUJn TOCm | Function |
|------------|------------|---|
| 0 | 0 | Toggle mode: Toggling proceeds when INTTAUJnIm is generated. |
| 0 | 1 | Setting prohibited |
| 1 | 0 | Synchronous channel operation mode: set when INT is generated on the master channel and reset when INT is generated on the slave channel. |
| 1 | 1 | Setting prohibited |

24.3.5.5 TAUJnTOL — TAUJn Channel Output Active Level Register

This register specifies the output logic of the channel output bit (TAUJnTO.TAUJnTOm).

Access: This register can be read/written in 8-bit units.

Address: <TAUJn_base> + 64_H

Value after reset: 00_H This register is initialized by various types of reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|------------|------------|------------|
| | — | — | — | — | TAUJnTOL03 | TAUJnTOL02 | TAUJnTOL01 | TAUJnTOL00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 24.31 TAUJnTOL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | TAUJnTOLm | Specify the output logic of the channel m output bit (TAUJnTO.TAUJnTOm): 0: Positive logic (active high) 1: Negative logic (active low) |

24.4 Operating Procedure

The following lists the general operation procedure for the TAUJn:

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUJnTTOUTm is also initialized and outputs a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJn function:
 - Set the operation mode
 - Set the channel output mode
 - Set any other control bits
3. Enable the counter by setting the TAUJnTS.TAUJnTSM bit to 1.
The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUJnTT.TAUJnTTm bit to 1. The counter can be forcibly restarted by setting the TAUJnTS.TAUJnTSM bit to 1.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions is given in the following sections:
 - **Section 24.12, Independent Channel Operation Functions**
 - **Section 24.13, Synchronous Channel Operation Functions**
2. Modify the function while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

24.5 Concepts of Synchronous Channel Operation

The synchronous channel operation depends on multiple channels and is affected by other channels. Therefore, several rules apply to the synchronous channel operation.

These rules are detailed in **Section 24.5.1, Rules of Synchronous Channel Operation**.

Two special features for the synchronous channel operation function are detailed in the following sections:

- **Section 24.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 24.6, Simultaneous Rewrite**

24.5.1 Rules of Synchronous Channel Operation

Number of master and slave channels

- Only even channels (CH0, CH2) can be set as master channels. Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.
Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master.
Example: If CH0 and CH2 are master channels, CH1 can be set as slave channels for CH0, but CH3 cannot.

Operating clock

- The same operating clock should be set for the master channel and the synchronized slave channel. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave usage and operating clocks are illustrated in **Figure 24.3**.

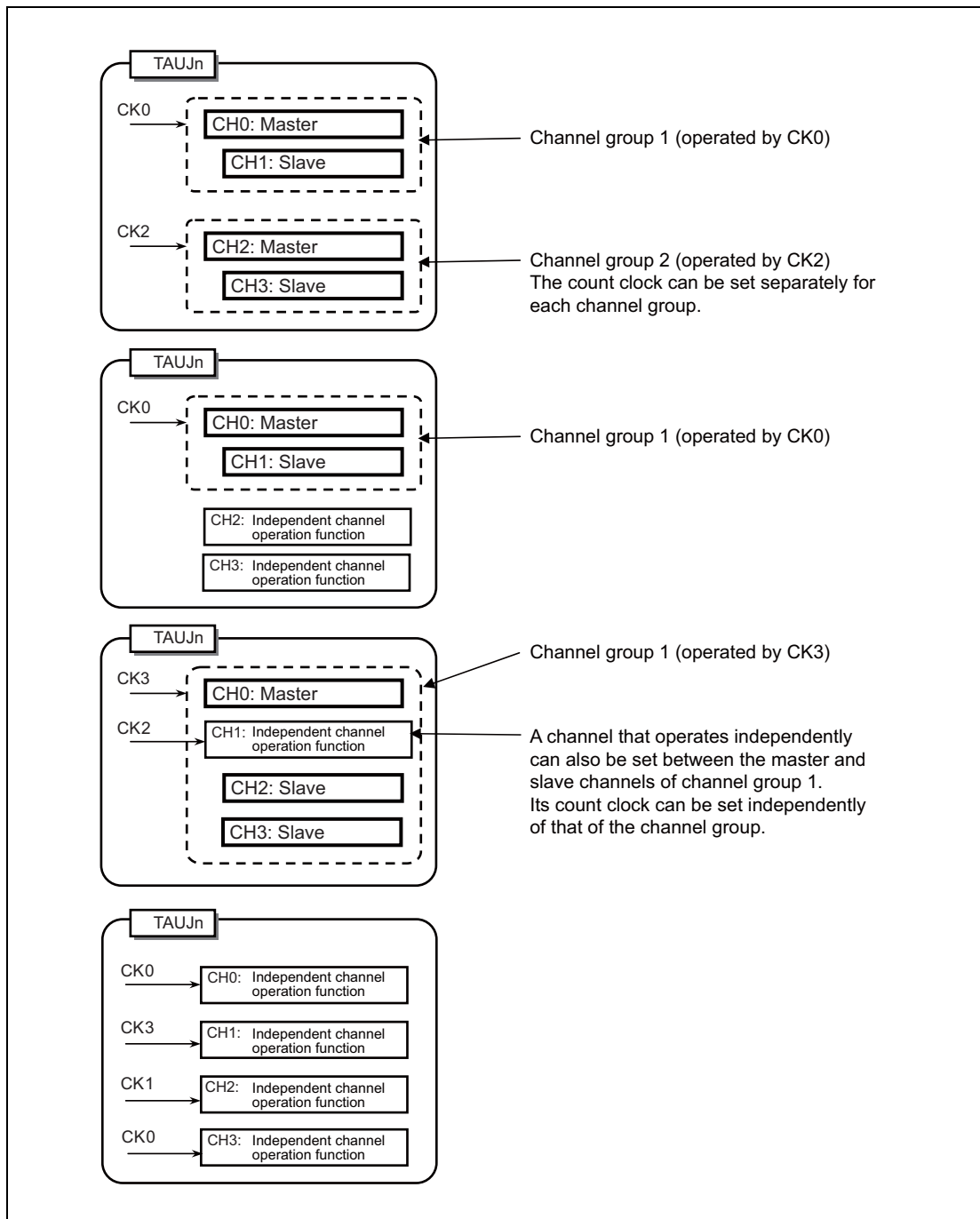


Figure 24.3 Grouping of Channels and Assignment of Count Clocks

24.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

24.5.2.1 Simultaneous Start and Stop within a TAUJ Unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSM bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTM bits of the channels should be set at the same time.

Setting 1 in the TAUJnTS.TAUJnTSM bits sets the corresponding TAUJnTE.TAUJnTEM bits to 1, enabling counting. The count start timing of the counter depends on the operating mode.

24.5.2.2 Simultaneous Start between TAUJ Units

Counters in different TAUJ units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

24.6 Simultaneous Rewrite

24.6.1 Functional Overview

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUJnCDRm and TAUJnTOL) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered when the counter on the master channel reaching a certain value.

Table 24.32 lists the simultaneous rewrite method (TAUJnRDM.TAUJnRDMm = 0).

Table 24.32 Simultaneous Rewrite Method

| Method | Simultaneous Rewrite Triggered Timing | TAUJnRDE.TAUJnRDEm |
|--------|--|--------------------|
| — | No simultaneous rewrite | 0 |
| A | The master channel (re)starts counting | 1 |

24.6.2 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

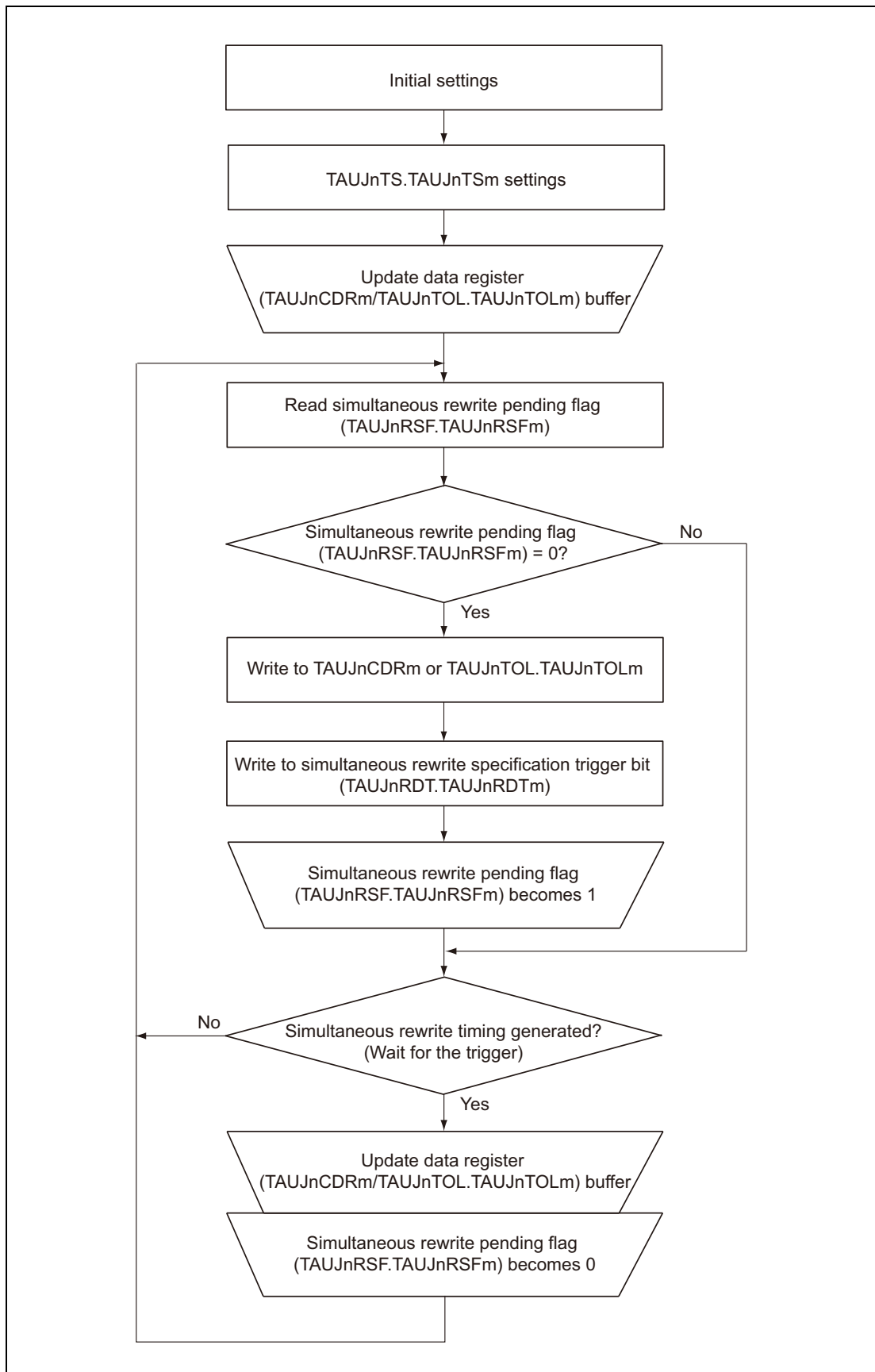


Figure 24.4 General Procedure for Simultaneous Rewrite

24.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set `TAUJnRDE.TAUJnRDEm = 1`

24.6.2.2 Start Counter and Count Operation

- To start all the `TAUJnCNTm` counters in the channel group, set the corresponding `TAUJnTS.TAUJnTSM` bits to 1. `TAUJnTOL.TAUJnTOLm` and the values in the data registers (`TAUJnCDRm`) are loaded into the corresponding `TAUJnTOL.TAUJnTOLm` buffer (`TAUJnTOL.TAUJnTOLm buf`) and data buffer registers (`TAUJnCDRm buf`) and the counters start.
- Setting the reload data trigger bit (`TAUJnRDT.TAUJnRDTm`) to 1 sets the reload flag (`TAUJnRSF.TAUJnRSFm`) to 1, enabling simultaneous rewrite. `TAUJnRSF.TAUJnRSFm` remains set to 1 until simultaneous rewrite is completed.
- When a specified trigger for simultaneous rewrite is detected, the `TAUJnRSF.TAUJnRSFm` bit is checked to see if simultaneous rewrite is enabled (`TAUJnRSF.TAUJnRSFm = 1`). If enabled, simultaneous rewrite is carried out. Otherwise simultaneous rewrite is not carried out and the system waits for detection of the next simultaneous rewrite trigger.

24.6.2.3 Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (`TAUJnRSF.TAUJnRSFm = 1`), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and the values are applied the next time the counter starts or restarts.
- When the simultaneous rewrite is completed, the `TAUJnRSF.TAUJnRSFm` bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

24.6.3 Other General Rules for Simultaneous Rewrite

The following rules also apply.

- `TAUJnRDE.TAUJnRDEm` and `TAUJnRDM.TAUJnRDMm` cannot be changed while the counter is in operation (`TAUJnTE.TAUJnTEm = 1`).
- `TAUJnTOL.TAUJnTOLm` can be rewritten only during operation using the PWM output function. For all other functions, `TAUJnTOL.TAUJnTOLm` should be written before the counter starts. If it is rewritten while any other function is used, `TAUJnTTOUTm` outputs an invalid wave.

24.6.4 Simultaneous Rewrite Procedure

The simultaneous rewrite procedure in PWM output mode is described in the following figure.

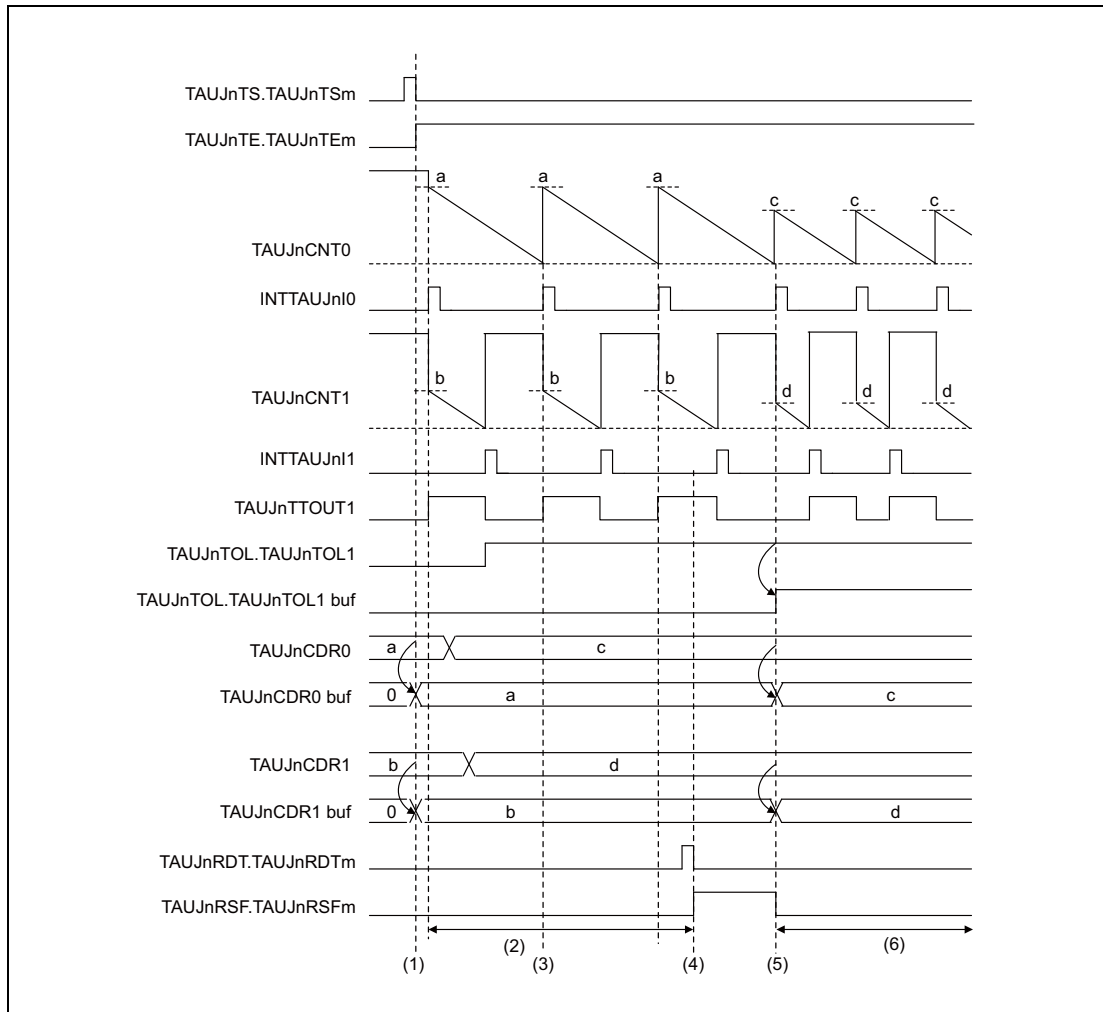


Figure 24.5 Simultaneous Rewrite in PWM Output Mode

Setting

CH0 is a master channel in PWM output mode, and CH1 represents an slave channel in PWM output mode. Simultaneous rewrite is applied when the master channel starts counting.

Description:

- (1) When $\text{TAUJnTS.TAUJnTSM} = 1$ is set, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer and the value of $\text{TAUJnTOL.TAUJnTOLm}$ is copied to the $\text{TAUJnTOL.TAUJnTOLm}$ buffer.
- (2) The TAUJnCDRm and $\text{TAUJnTOL.TAUJnTOLm}$ registers can be written at any time.
- (3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled ($\text{TAUJnRSF.TAUJnRSFm} = 0$).
- (4) The reload data trigger bit ($\text{TAUJnRDT.TAUJnRDTm}$) is set to 1 which sets the status flag ($\text{TAUJnRSF.TAUJnRSFm} = 1$), enabling simultaneous rewrite.
- (5) Simultaneous rewrite is triggered when CH0 restarts counting, because simultaneous rewrite is enabled. The TAUJnCDRm value is loaded into the TAUJnCDRm buffer and the $\text{TAUJnTOL.TAUJnTOLm}$ value is loaded into the $\text{TAUJnTOL.TAUJnTOLm}$ buffer.
- (6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of TAUJnCDRm and $\text{TAUJnTOL.TAUJnTOLm}$ can be changed again.

24.7 Channel Output Modes

The output of the TAUJnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUJnTOE.TAUJnTOEm = 0)
When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOM) is sent out of the output pin (TAUJnTTOUTm).
- By TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)
When controlled by TAUJ signals, the output level of TAUJnTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOM is updated accordingly to reflect the value of TAUJnTTOUTm.
 - Independently (TAUJnTOM.TAUJnTOMm = 0)
In case of independent operation, the output of the TAUJnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUJnTOM.TAUJnTOMm = 0).
 - Synchronously (TAUJnTOM.TAUJnTOMm = 1)
In case of synchronous operation, the output of the TAUJnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOM bit can always be read to determine the current value of TAUJnTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

(1) Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 24.33, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 24.7.2, Channel Output Modes Controlled Independently by TAUJn Signals**
- **Section 24.7.3, Channel Output Modes Controlled Synchronously by TAUJn Signals**

(2) Batch operation of TAUJnTOM bit

Whether a set value is reflected to the TAUJnTO.TAUJnTOM bit or not is controlled by the TAUJnTOE.TAUJnTOEm bit.

The TAUJnTO.TAUJnTOM setting is written only to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 0 when a write to the TAUJnTO register is attempted.

No TAUJnTO.TAUJnTOM setting is reflected to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 1.

NOTE

TAUJnTO.TAUJnTOM bit is placed so that its bit number corresponds to a channel number.

(3) Output logic

Positive logic or negative logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of TAUJnTOL.TAUJnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function. Otherwise, changes to TAUJnTOL.TAUJnTOLm result in an undefined TAUJnTTOUTm signal output.

See **Section 24.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in the following table (TAUJnTOC.TAUJnTOCm = 0).

Table 24.33 Channel Output Modes

| Channel Output Mode | TAUJnTOE. TAUJnTOEm | TAUJnTOM. TAUJnTOMm |
|--|------------------------|------------------------|
| By software | | |
| Independent channel output mode controlled by software | 0 | x |
| By TAUJ signals, independently | | |
| Independent channel output mode 1 | 1 | 0 |
| By TAUJ signals, synchronously | | |
| Synchronous channel output mode 1 | 1 | 1 |

- Any combination not in the list is prohibited.
- Any value can be set in the bit indicated as “x”.

NOTE

Do not modify the bits listed below during counting operation (TAUJnTE.TAUJnTEm = 1).

- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

24.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUJnTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

- (1) Set TAUJnTO.TAUJnTOm to specify the initial level of the TAUJnTTOUTm output.
- (2) Set channel output mode according to Table 24.33, Channel Output Modes, and the output logic using the TAUJnTOL.TAUJnTOLm bit.
- (3) Start the counter (TAUJnTS.TAUJnTSm = 1).

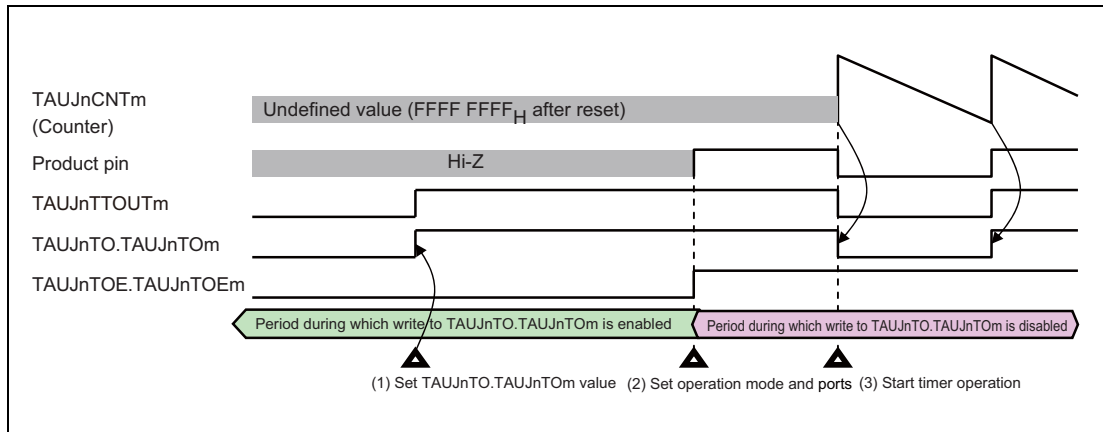


Figure 24.6 General Procedure for Specifying a TAUJnTTOUTm Channel Output Mode

24.7.2 Channel Output Modes Controlled Independently by TAUJn Signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in Table 24.33, Channel Output Modes.

(1) Independent channel output mode 1

Set/reset conditions

In this output mode, TAUJnTTOUTm toggles when INTTAUJnIm is detected.

The value of TAUJnTOL.TAUJnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in Table 24.33, Channel Output Modes.

24.7.3 Channel Output Modes Controlled Synchronously by TAUJn Signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in **Table 24.33, Channel Output Modes**.

(1) Synchronous channel output mode 1

Set/reset conditions

In this output mode, INTTAUJnIm of master channel serves as a set signal and INTTAUJnIm of the slave channel as a reset signal. If INTTAUJnIm of the master channel and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of the master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 24.33, Channel Output Modes**.

24.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUJnTS.TAUJnTSM is set to 1 in each operating mode.

The value of data register (TAUJnCDRm register) and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

24.8.1 Interval Timer Mode, Capture Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUJnTS.TAUJnTSM is set to 1. The value of data register is also loaded when the counter starts.

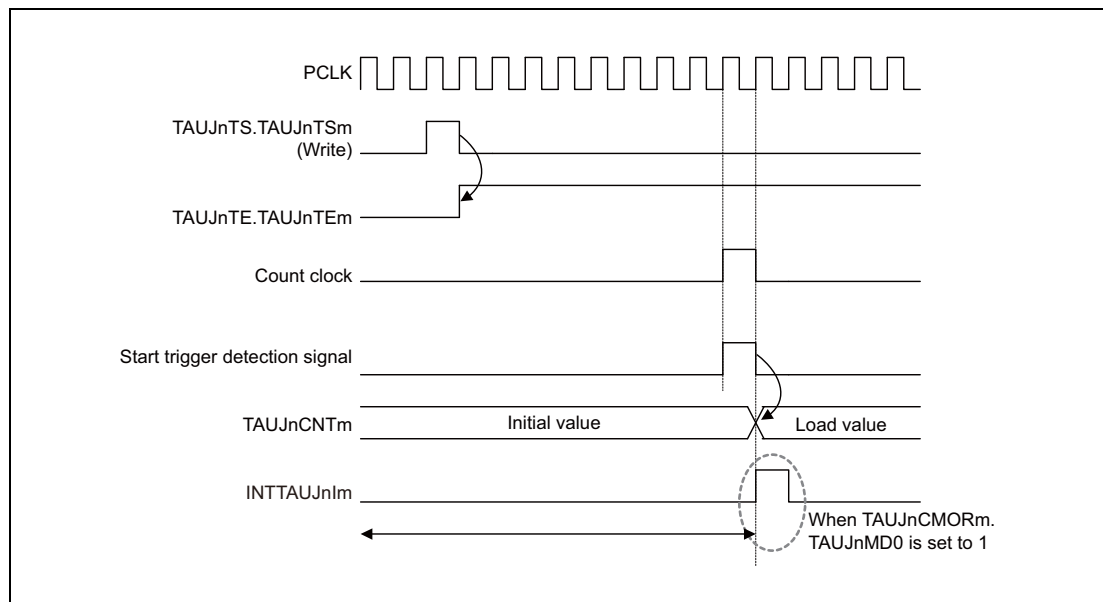


Figure 24.7 Start Timing in Interval Timer Mode, Capture Mode, and Count Capture Mode

24.8.2 Other Operating Modes

In other operating modes, count clock cycle is irrelevant to start of counter operation. The counter is triggered only by TAUJnTTINm valid edge detection. As the counter starts, the value of the data register is also loaded. The cycle of the counter clock is irrelevant to the start of counter operation but does determine the frequency with which all operations take place.

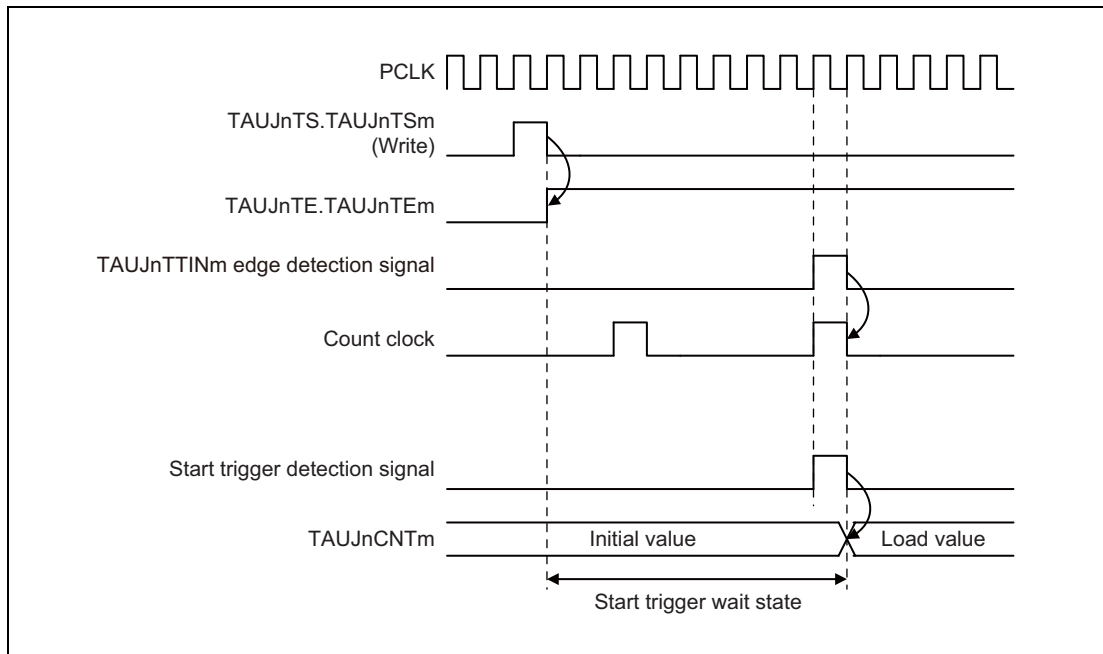


Figure 24.8 Count Start Timing in Other Operating Modes

24.9 TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts

The TAUJnCMORm.TAUJnMD0 bits are used to specify whether the corresponding INTTAUJnIm interrupts are generated when the counter starts. The effect of the TAUJnCMORm.TAUJnMD0 bits on INTTAUJnIm generation when counting starts and on TAUJnTTOUTm depends on the selected mode. For details, see the descriptions of TAUJnCMORm.TAUJnMD0 in sections on the related modules.

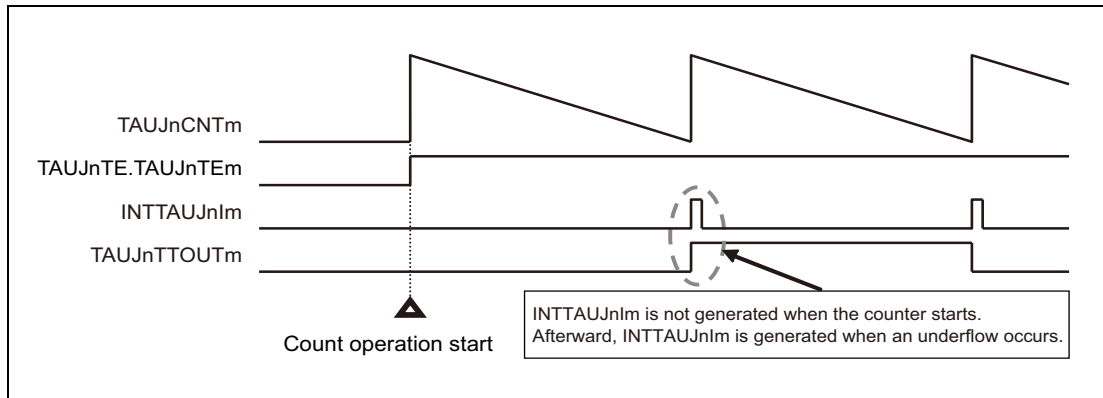


Figure 24.9 INTTAUJnIm Generation Timing (TAUJnCMORm.TAUJnMD0 = 0)

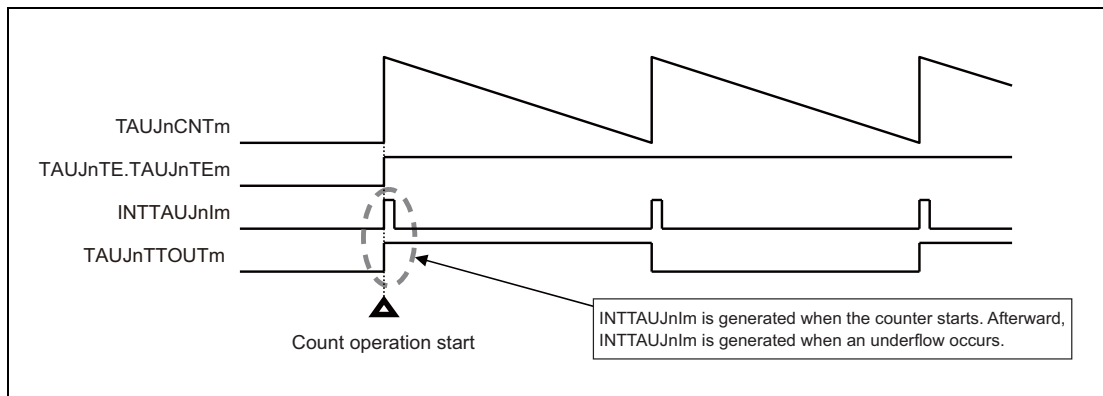


Figure 24.10 INTTAUJnIm Generation Timing (TAUJnCMORm.TAUJnMD0 = 1)

24.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches FFFF FFFF_H and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operations in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000 0000_H at the same time as the first channel overflows (TAUJnCNTm = FFFF FFFF_H).
- Set TAUJnCDRm of the second channel to FFFF FFFF_H.
- The two channels must count at the same speed (i.e. they must have the same count clock).

Result:

The down-counter of the second channel reaches 0000 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUJnCNTm = FFFF FFFF_H). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

24.10.1 Combination of the TAUJnTTINm Input Position Detection Function and the Interval Timer Function

When the counters of both channels are started simultaneously, INTTAUJnIm of the interval timer function can detect the overflow when TAUJnCNTm of the TAUJnTTINm input position detection function exceeds FFFF FFFF_H.

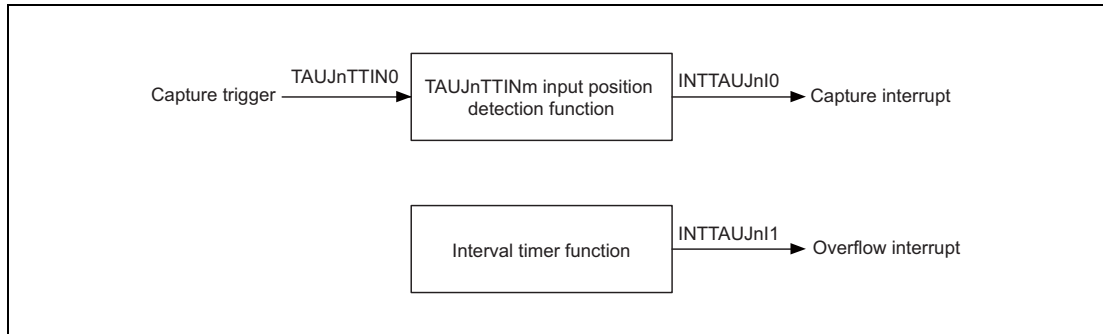


Figure 24.11 Example of Combination of the TAUJnTTINm Input Position Detection Function and the Interval Timer Function (Ch0, Ch1)

Timing diagram

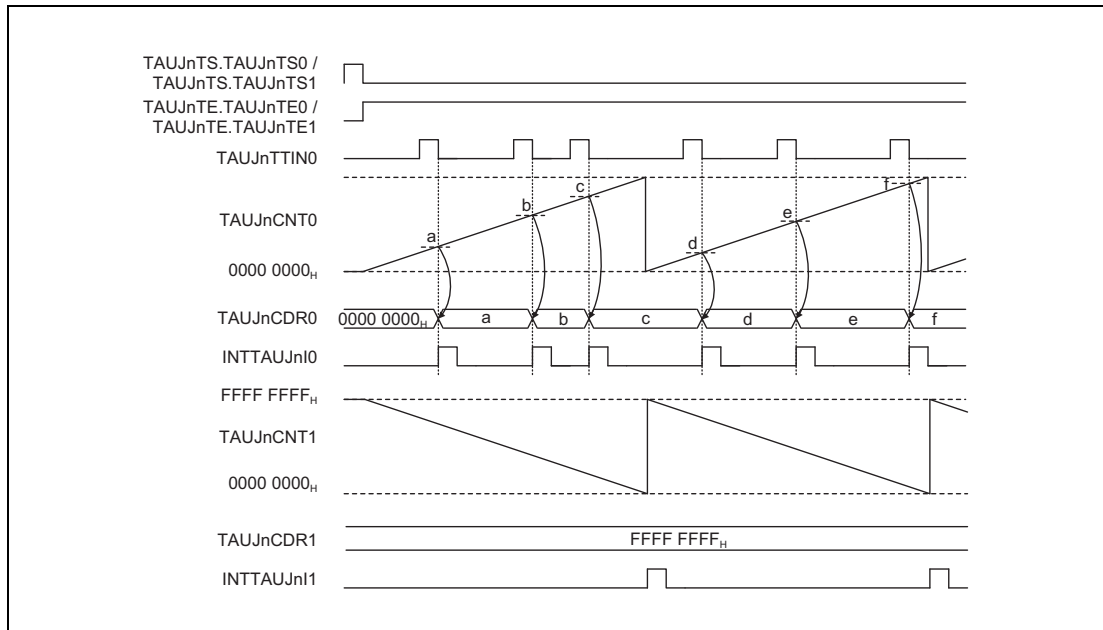


Figure 24.12 Example of Interrupt Generation via Combination of the TAUJnTTINm Input Position Detection Function and the Interval Timer Function (Ch0, Ch1)

24.11 TAUJnTTINm Edge Detection

Edge detection is based on the operating clock. This means that an edge can only be detected at the next rising edge of the operating clock. This can lead to a maximum delay of one operating clock cycle.

The following figure shows when edge detection takes place.

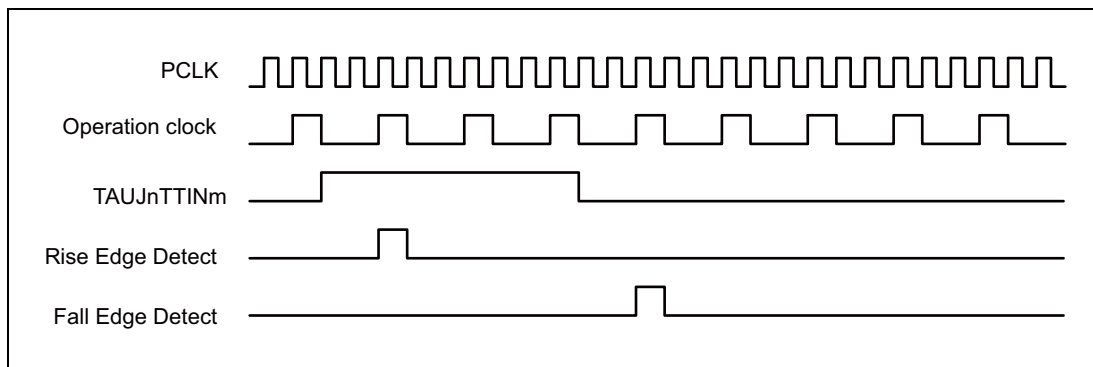


Figure 24.13 Basic Edge Detection Timing

Figure 24.13 shows an image of operation timing. In the actual operation, delay time occurs due to noise filter and synchronization circuit between the TAUJnIm pin and TAUJn.

- Delay time by a noise filter + delay time for edge detection (maximum one sampling clock cycle)

24.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUJ. For a general overview of independent channel operation functions, see **Section 24.2, Overview**.

24.12.1 Interval Timer Function

24.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode, refer to **Table 24.34, Contents of the TAUJnCMORm register for the Interval Timer Function**.
- The channel output mode must be set to Independent Channel Output Mode 1, refer to **Section 24.7, Channel Output Modes**.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSM) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation. The current value of TAUJnCDRm is written to TAUJnCNTm and the counter starts to count down from this value.

When the counter reaches 0000 0000_H, INTTAUJnIm is generated and the TAUJnTTOUTm signal toggles. TAUJnCNTm then loads the TAUJnCDRm value and subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEM to 0. TAUJnCNTm and TAUJnTTOUTm stop but retain their values. The counter can be restarted by setting TAUJnTS.TAUJnTSM to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSM to 1 during operation.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJnTTOUTm does not toggle. This results in an inverted TAUJnTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1. For details refer to **Section 24.9, TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts**.

24.12.1.2 Equations

$$\text{INTTAUJnIm cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm} + 1)$$

$$\text{TAUJnTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm} + 1) \times 2$$

24.12.1.3 Block Diagram and General Timing Diagram

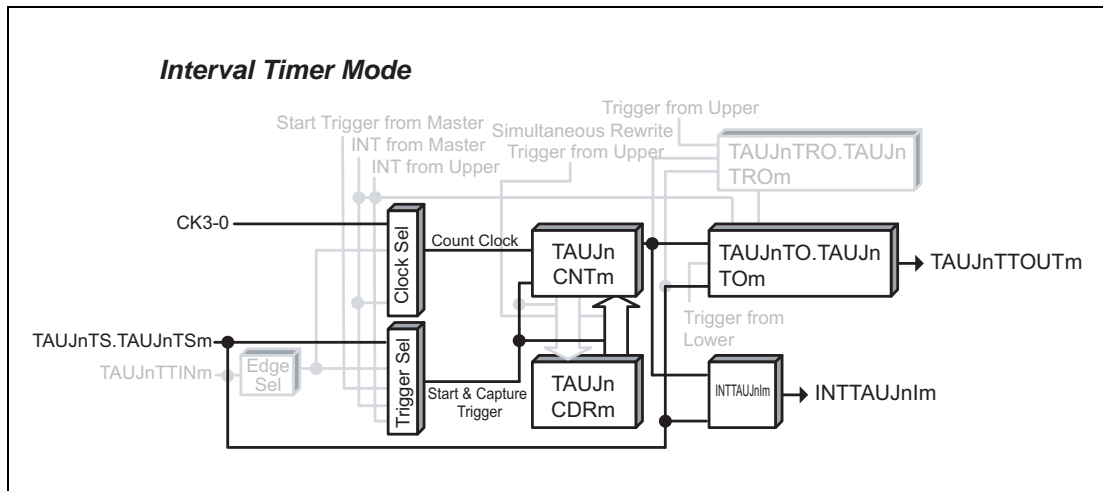


Figure 24.14 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram:

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1)

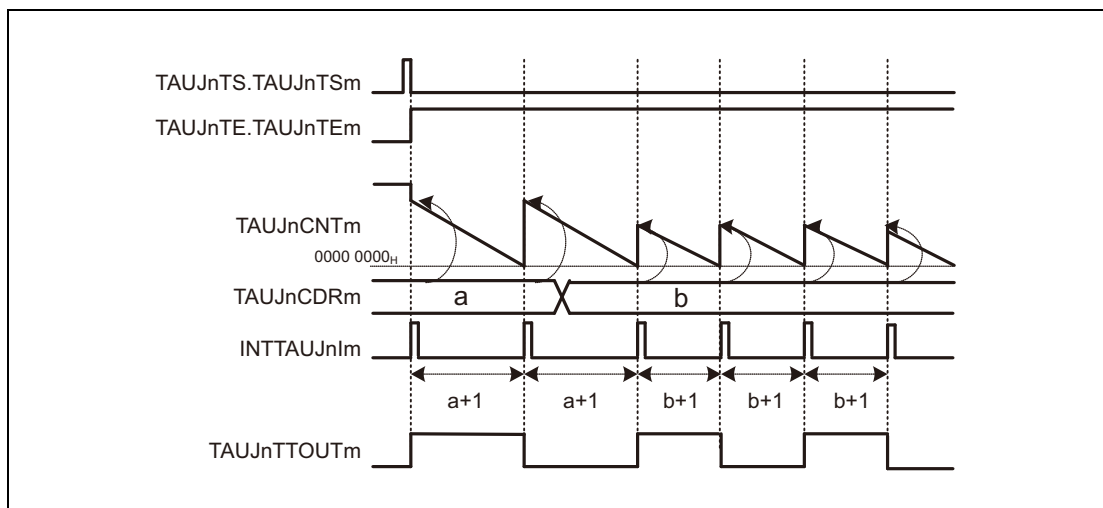


Figure 24.15 General Timing Diagram for Interval Timer Function

24.12.1.4 Register Settings

(1) TAUJnCMORM

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.34 Contents of the TAUJnCMORM register for the Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 0 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 000 _B . |
| 7, 6 | TAUJnCOS[1:0] | Write 00 _B . |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 0000 _B . |
| 0 | TAUJnMD0 | 0: INTTAUJnIm not generated and TAUJnTTOUTm does not toggle at operation start or restart 1: Generates INTTAUJnIm and toggles TAUJnTTOUTm at operation start or restart |

(2) TAUJnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.35 Contents of the TAUJnCMURm register for the Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 00: Not used, so set to 00 |

(3) Channel output mode**Table 24.36 Control Bit Settings for Independent Channel Output Mode 1**

| Bit name | Setting |
|--------------------|------------------------|
| TAUJnTOE.TAUJnTOEm | Write 1 _B . |
| TAUJnTOM.TAUJnTOMm | Write 0 _B . |
| TAUJnTOC.TAUJnTOCm | Write 0 _B . |
| TAUJnTOL.TAUJnTOLm | Write 0 _B . |

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTOUTm can then be controlled independently of the interrupts. For details refer to **Section 24.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Interval Timer Function. Therefore, these registers must be set to 0.

Table 24.37 Simultaneous Rewrite Settings for Interval Timer Function

| Bit name | Setting |
|--------------------|--|
| TAUJnRDE.TAUJnRDEm | 0: Disables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0 |

24.12.1.5 Operating Procedure for Interval Timer Function

Table 24.38 Operating Procedure for Interval Timer Function

| | Operation | Status of TAUJn |
|-------------------------|--|--|
| Initial channel setting | Set the TAUJnCMORm register and TAUJnCMURm registers as described in Table 24.34, Contents of the TAUJnCMORm register for the Interval Timer Function and Table 24.35, Contents of the TAUJnCMURm register for the Interval Timer Function . | Channel operation is stopped. |
| | Set the value of the TAUJnCDRm register. | |
| | Set the channel output mode by setting the control bits as described in Table 24.36, Control Bit Settings for Independent Channel Output Mode 1 . | |
| Restart | Start operation Set TAUJnTS.TAUJnTSM to 1. TAUJnTS.TAUJnTSM is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEM is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles. |
| | During operation The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times. | TAUJnCNTm counts down. When the counter reaches 0000 0000 _H : <ul style="list-style-type: none"> • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation • INTTAUJnIm is generated and TAUJnTTOUTm toggles. |
| Stop operation | Set TAUJnTT.TAUJnTTM to 1. TAUJnTT.TAUJnTTM is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEM is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values. |

24.12.1.6 Specific Timing Diagrams

(1) TAUJnCDRm = 0000 0000_H, count clock = PCLK/2

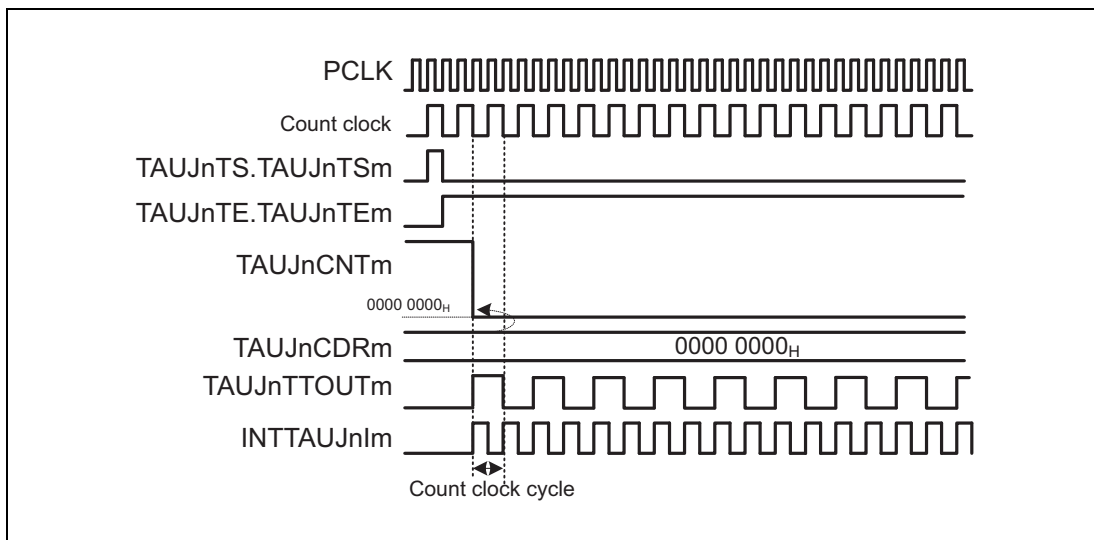


Figure 24.16 TAUJnCDRm = 0000 0000_H, Count Clock = PCLK/2

- If TAUJnCDRm = 0000 0000_H and the count clock = PCLK/2, the TAUJnCDRm value is written to TAUJnCNTm every count clock, meaning that TAUJnCNTm is always 0000 0000_H.
- INTTAUJnIm is generated every count clock, resulting in TAUJnTTOUTm toggling every count clock.

(2) TAUJnCDRm = 0000 0000_H, count clock = PCLK

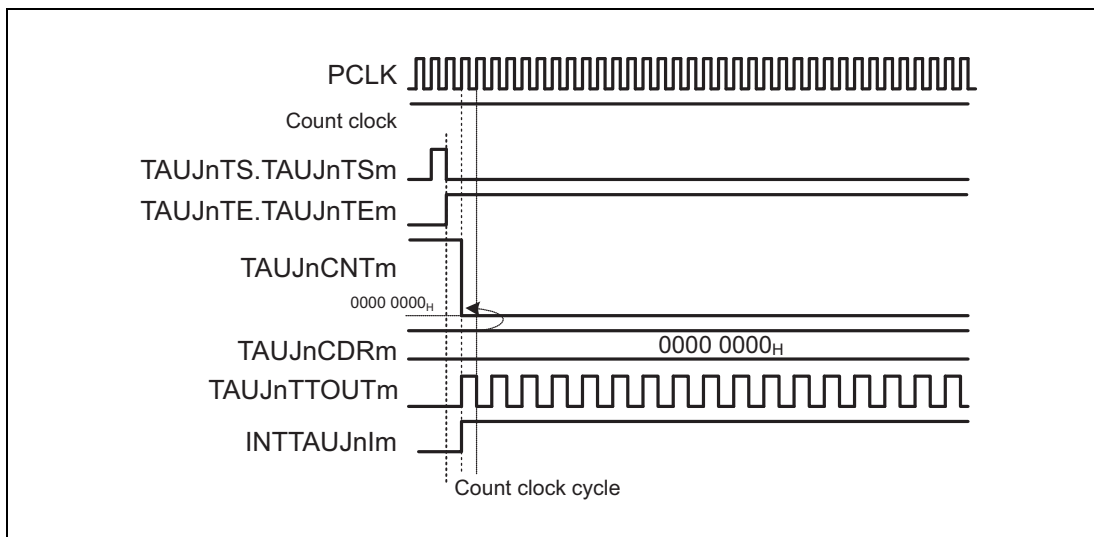


Figure 24.17 TAUJnCDRm = 0000 0000_H, Count Clock = PCLK

- If TAUJnCDRm = 0000 0000_H and the count clock = PCLK, the TAUJnCDRm value is written to TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always 0000 0000_H.
- INTTAUJnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated. TAUJnTTOUTm is toggled every PCLK clock.

(3) Operation stop and restart

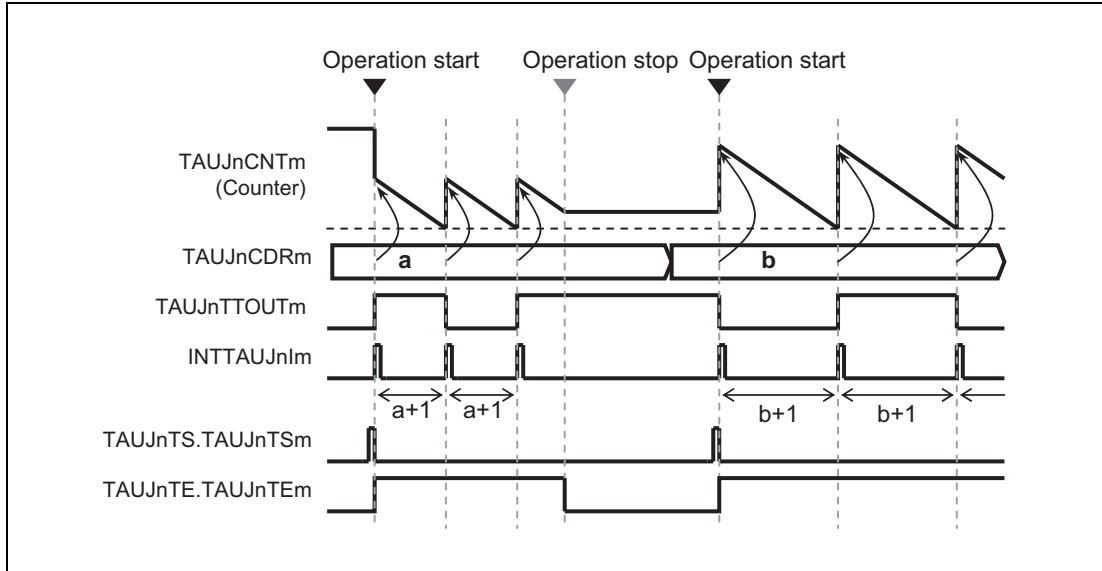


Figure 24.18 Operation Stop and Restart, TAUJnCMORm.TAUJnMD0 = 1

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

(4) Forced restart

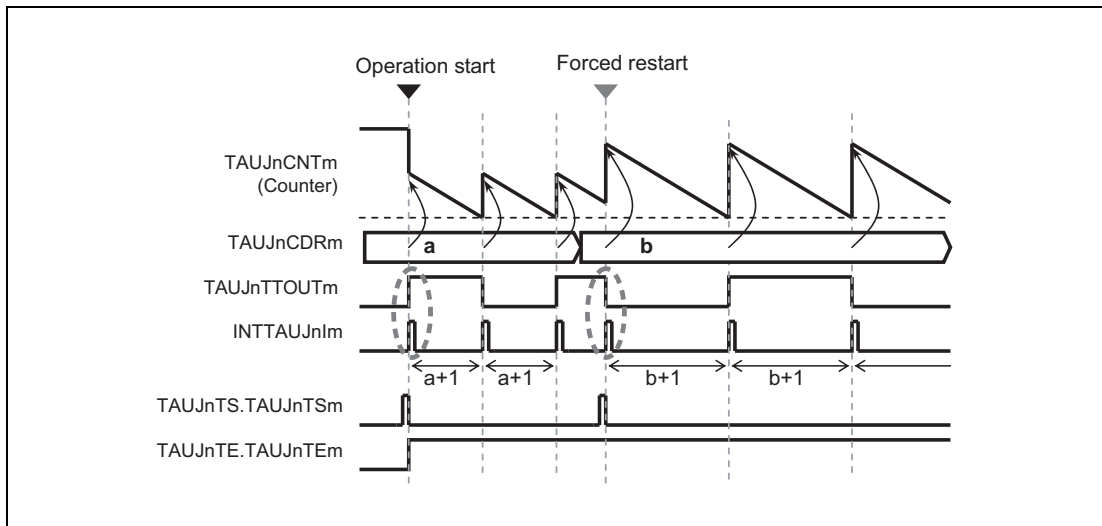


Figure 24.19 Forced Restart Operation, TAUJnCMORm.TAUJnMD0 = 1

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- At the time of a forced restart, the value of TAUJnCDRm is reflected to TAUJnCNTm and counting starts. Perform a forced restart to reflect the modified value of TAUJnCDRm immediately.

24.12.2 TAUJnTTINm Input Interval Timer Function

24.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJnTTINm input edge is detected. When an interrupt is generated, the TAUJnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode, refer to **Table 24.39, Contents of the TAUJnCMORm register for TAUJnTTINm Input Interval Timer Function**.
- The channel output mode must be set to Independent Channel Output Mode 1, refer to **Section 24.7, Channel Output Modes**.

Description

This function operates in an identical manner to the Interval Timer Function (see **Section 24.12.1, Interval Timer Function**), except that this function is restarted by a valid TAUJnTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm.TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edge can be selected.

24.12.2.2 Equations

$$\text{INTTAUJnIm cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm} + 1)$$

$$\text{TAUJnTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm} + 1) \times 2$$

24.12.2.3 Block Diagram and General Timing Diagram

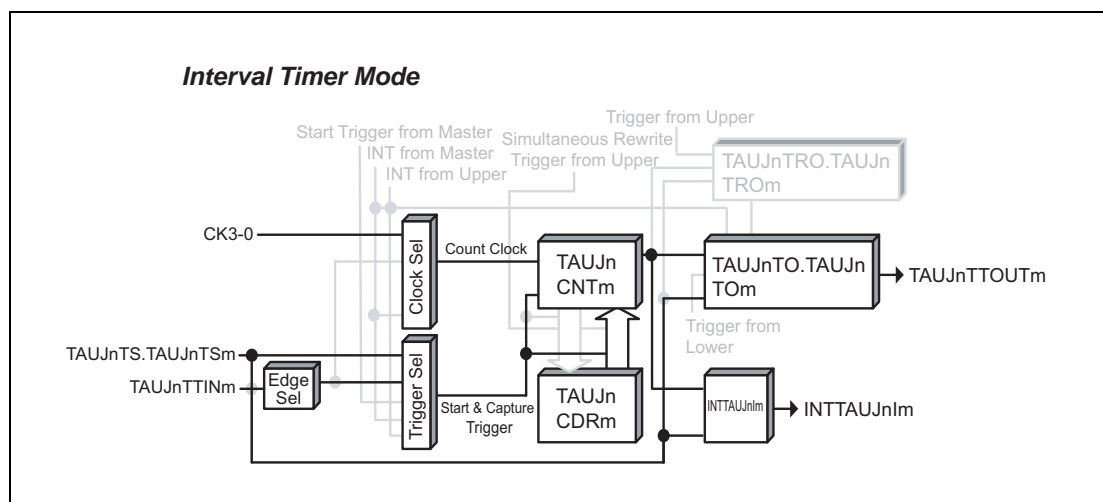


Figure 24.20 Block Diagram for TAUJnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram:

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

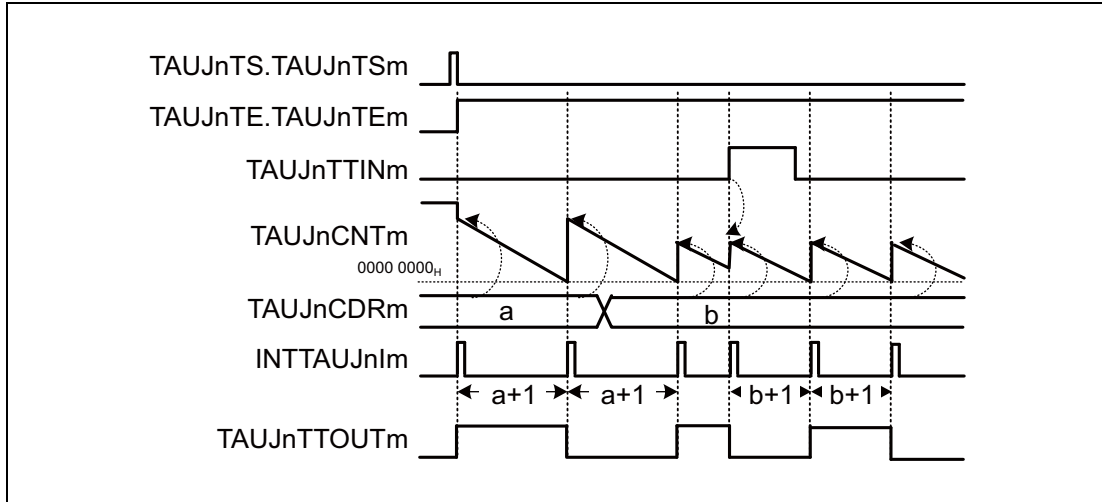


Figure 24.21 General Timing Diagram for TAUJnTTINm Input Interval Timer Function

24.12.2.4 Register Settings

(1) TAUJnCMORM

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMDO |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.39 Contents of the TAUJnCMORM register for TAUJnTTINm Input Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 0 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 001 _B . |
| 7, 6 | TAUJnCOS[1:0] | Write 00 _B . |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 0000 _B . |
| 0 | TAUJnMDO | 0: INTTAUJnIm not generated and TAUJnTTOUTm does not toggle at operation start 1: Generates INTTAUJnIm and toggles TAUJnTTOUTm at operation start |

(2) TAUJnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.40 Contents of the TAUJnCMURm register for TAUJnTTINm Input Interval Timer Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited |

(3) Channel output mode**Table 24.41 Control Bit Settings for Independent Channel Output Mode 1**

| Bit name | Setting |
|--------------------|------------------------|
| TAUJnTOE.TAUJnTOEm | Write 1 _B . |
| TAUJnTOM.TAUJnTOMm | Write 0 _B . |
| TAUJnTOC.TAUJnTOCm | Write 0 _B . |
| TAUJnTOL.TAUJnTOLm | Write 0 _B . |

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTOUTm can then be controlled independently of the interrupts. For details refer to **Section 24.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

Table 24.42 Simultaneous Rewrite Settings for TAUJnTTINm Input Interval Timer Function

| Bit name | Setting |
|--------------------|--|
| TAUJnRDE.TAUJnRDEm | 0: Disables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0 |

24.12.2.5 Operating Procedure for TAUJnTTINm Input Interval Timer Function

Table 24.43 Operating Procedure for TAUJnTTINm Input Interval Timer Function

| | Operation | Status of TAUJn |
|-------------------------|--|---|
| Initial channel setting | Set the TAUJnCMORm register and TAUJnCMURm registers as described in Table 24.39, Contents of the TAUJnCMORm register for TAUJnTTINm Input Interval Timer Function and Table 24.40, Contents of the TAUJnCMURm register for TAUJnTTINm Input Interval Timer Function . | Channel operation is stopped. |
| | Set the value of the TAUJnCDRm register | |
| | Set the channel output mode by setting the control bits as described in Table 24.41, Control Bit Settings for Independent Channel Output Mode 1 . | |
| Start operation | Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCnTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles |
| | <p>The values of the TAUJnCMURm.TAUJnTIS[1:0] and TAUJnCDRm registers can be changed at any time. The TAUJnCnTm register can be read at all times.</p> <p>Detection of TAUJnTTINm edge</p> | <p>TAUJnCnTm counts down. When the counter reaches 0000 0000_H:</p> <ul style="list-style-type: none"> TAUJnCnTm reloads the TAUJnCDRm value and continues count operation INTTAUJnIm is generated and TAUJnTTOUTm toggles <p>When a TAUJnTTINm input valid edge is detected during count operation, TAUJnCnTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.</p> |
| Stop operation | Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCnTm and TAUJnTTOUTm stop and retain their current values. |

Restart

During operation

Stop operation

24.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 24.12.1, Interval Timer Function** also apply, and in addition, the counter can also be restarted by a valid TAUJnTTINm input edge.

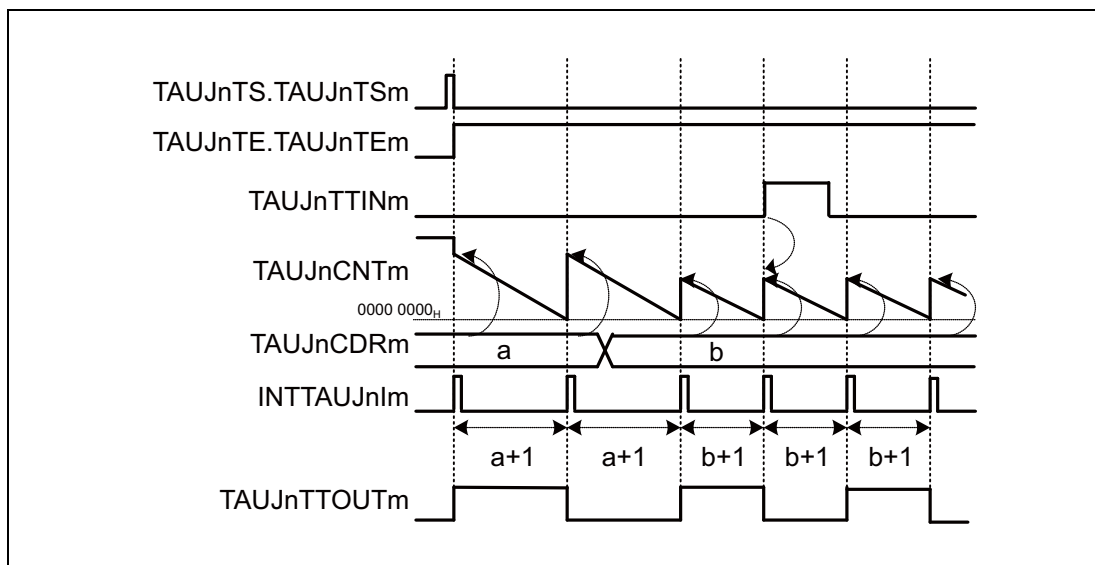


Figure 24.22 Counter Triggered by Rising TAUJnTTINm Input Edge
 (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORM.TAUJnMD0 = 1

If an effective TAUJnTTINm input edge is detected, an interrupt is generated which causes TAUJnTTOUTm to toggle. In this example, the effective edge is a rising edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B).

24.12.3 TAUJnTTINm Input Pulse Interval Measurement Function

24.12.3.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJnTTINm input signals.

Prerequisites

- The operation mode must be set to capture mode, refer to **Table 24.45, Contents of the TAUJnCMORm register for TAUJnTTINm Input Pulse Interval Measurement Function**.
- TAUJnTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter resets to 0000 0000_H and subsequently continues operation.

If the counter reaches FFFF FFFF_H before a valid TAUJnTTINm edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0]:

Table 24.44 Effects of an Overflow

| TAUJnCMORm. TAUJnCOS[1:0] | When overflow occurs | | When a valid TAUJnTTINm input is then detected | |
|------------------------------|----------------------------------|------------------------|--|------------------------|
| | TAUJnCDRm | TAUJnCSRm. TAUJnOVF | TAUJnCDRm and TAUJnCNTm | TAUJnCSRm. TAUJnOVF |
| 00 | Unchanged | 0 | TAUJnCNTm written to TAUJnCDRm | 1 |
| 01 | | 1 | | |
| 10 | Set to FFFF FFFF _H | 0 | TAUJnCNTm set to 0, TAUJnCDRm unchanged | Unchanged |
| 11 | | 1 | | |

If an overflow is set (TAUJnCSRm.TAUJnOVF = 1) when TAUJnCMORm.TAUJnCOS[0] = 1, it can only be cleared by setting TAUJnCSCm.TAUJnCLOV = 1.

The combination of the value of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJnTTINm signal. However, if an overflow occurs multiple times before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEm = 0. TAUJnCNTm stops but retains its value. While the function is stopped, TAUJnTTINm input valid edge detection and TAUJnCNTm capture are not performed.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details refer to **Section 24.9, TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts.**

NOTE

When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not written to TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

24.12.3.2 Equations

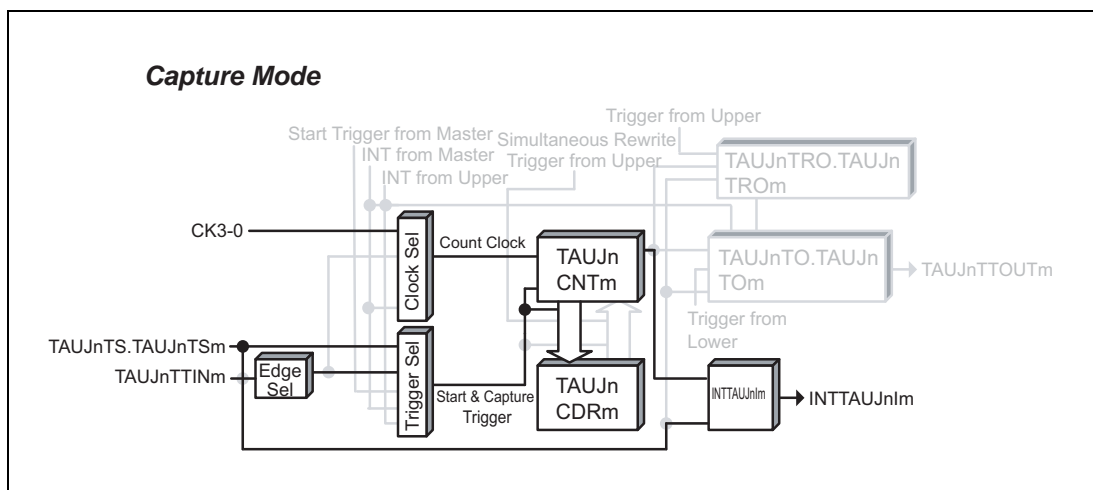
$$\text{TAUJnTTINm input pulse interval} = \text{count clock cycle} \times [(\text{TAUJnCSRm.TAUJnOVF} \times (\text{FFFF FFFF}_H + 1)) + \text{TAUJnCDRm capture value} + 1]$$
24.12.3.3 Block Diagram and General Timing Diagram

Figure 24.23 Block Diagram for TAUJnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram:

- INTTAUJnIm not generated at operation start (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)
- When a valid TAUJnTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B)

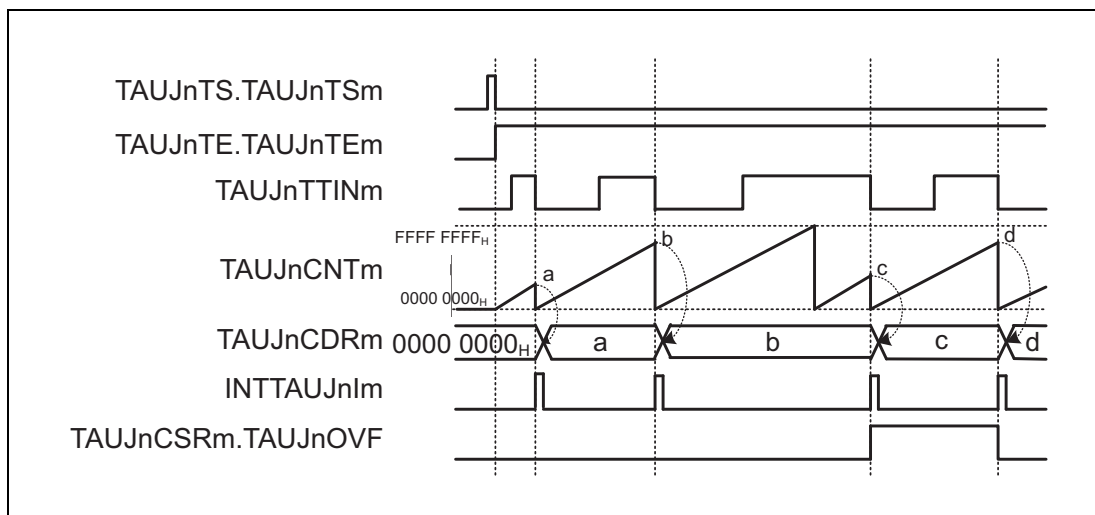


Figure 24.24 General Timing Diagram for TAUJnTTINm Input Pulse Interval Measurement Function

24.12.3.4 Register Settings

(1) TAUJnCMORM

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.45 Contents of the TAUJnCMORM register for TAUJnTTINm Input Pulse Interval Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 0 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 001 _B . |
| 7, 6 | TAUJnCOS[1:0] | See Table 24.44, Effects of an Overflow |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 0010 _B . |
| 0 | TAUJnMD0 | 0: INTTAUJnIm not generated at operation start 1: Generates INTTAUJnIm at operation start |

(2) TAUJnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.46 Contents of the TAUJnCMURm register for TAUJnTTINm Input Pulse Interval Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited |

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Pulse Interval Measurement Function. Therefore, these registers must be set to 0.

Table 24.47 Simultaneous Rewrite Settings for TAUJnTTINm Input Pulse Interval Measurement Function

| Bit name | Setting |
|--------------------|--|
| TAUJnRDE.TAUJnRDEm | 0: Disables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0 |

24.12.3.5 Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function

Table 24.48 Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function

| | Operation | Status of TAUJn |
|-------------------------|--|--|
| Initial channel setting | Set the TAUJnCMORm register and TAUJnCMURm registers as described in Table 24.45, Contents of the TAUJnCMORm register for TAUJnTTINm Input Pulse Interval Measurement Function and Table 24.46, Contents of the TAUJnCMURm register for TAUJnTTINm Input Pulse Interval Measurement Function . | Channel operation is stopped. |
| | The TAUJnCDRm register operates as a capture register. | |
| Restart | Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCnTm is cleared to 0000 0000 _H . INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1. |
| | Detection of TAUJnTTINm edges. The TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. TAUJnCSCm.TAUJnCLOV bit can be written to 1. (TAUJnCSRm.TAUJnOVF bit is cleared to 0.) | TAUJnCnTm starts to count up from 0000 0000 _H . When a TAUJnTTINm valid edge is detected: <ul style="list-style-type: none"> TAUJnCnTm transfers (captures) its value to TAUJnCDRm, and returns to 0000 0000_H INTTAUJnIm is then generated. Afterwards, this procedure is repeated. |
| Stop operation | Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCnTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values. |

24.12.3.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

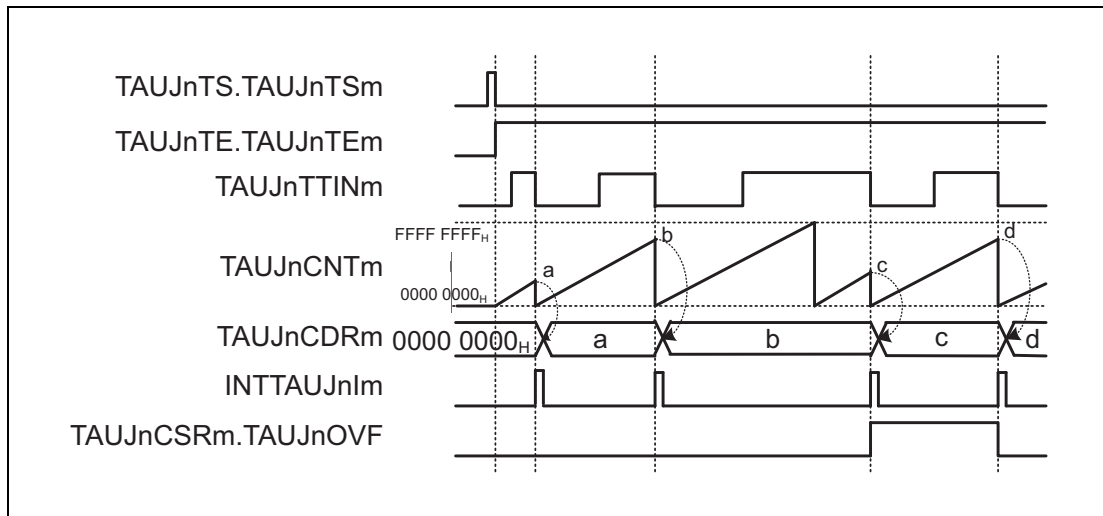


Figure 24.25 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF remains = 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is written to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- If the next valid TAUJnTTINm input edge is detected when no overflow occurs, TAUJnCSRm.TAUJnOVF is cleared to 0.

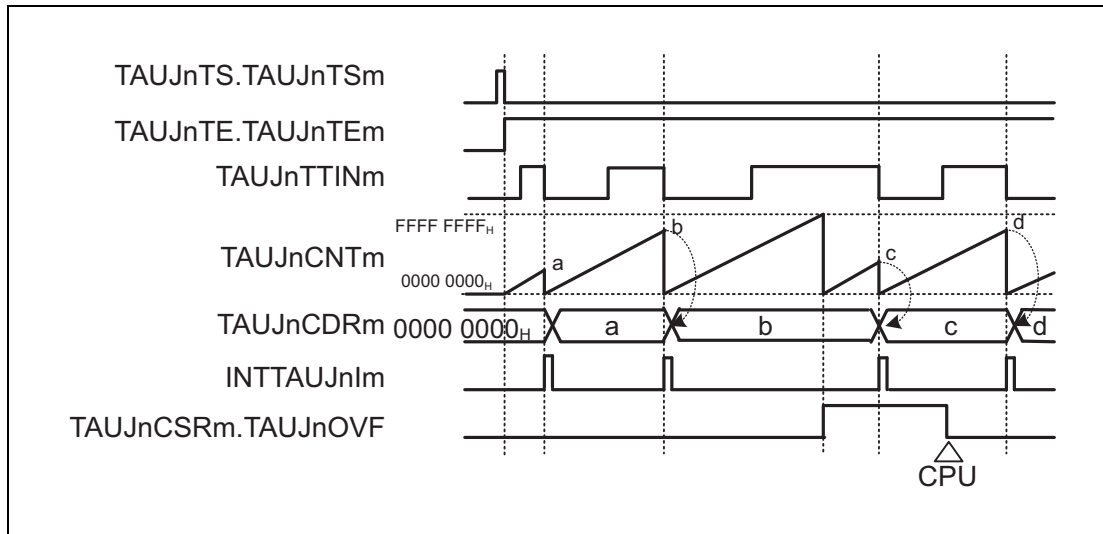
(2) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

Figure 24.26 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is written to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

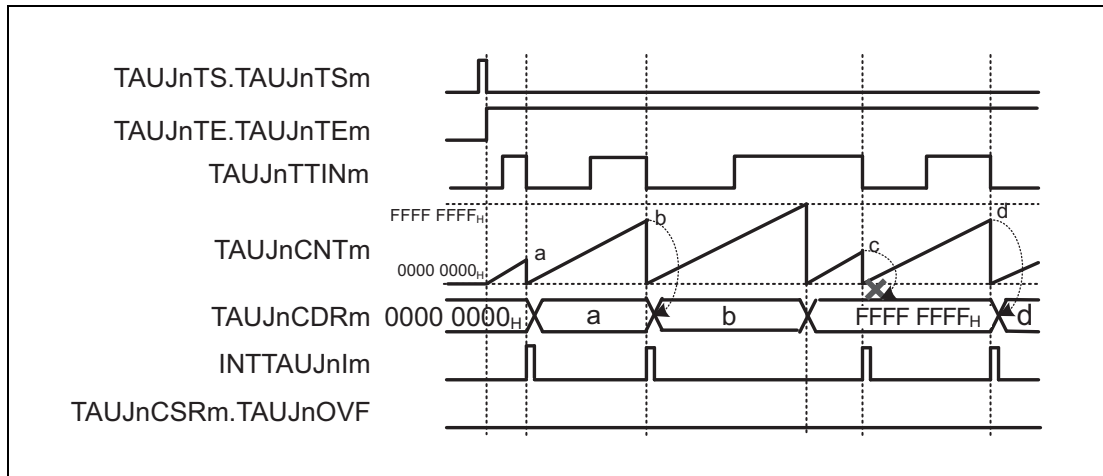
(3) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

Figure 24.27 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF_H and TAUJnCSRm.TAUJnOVF remains = 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.

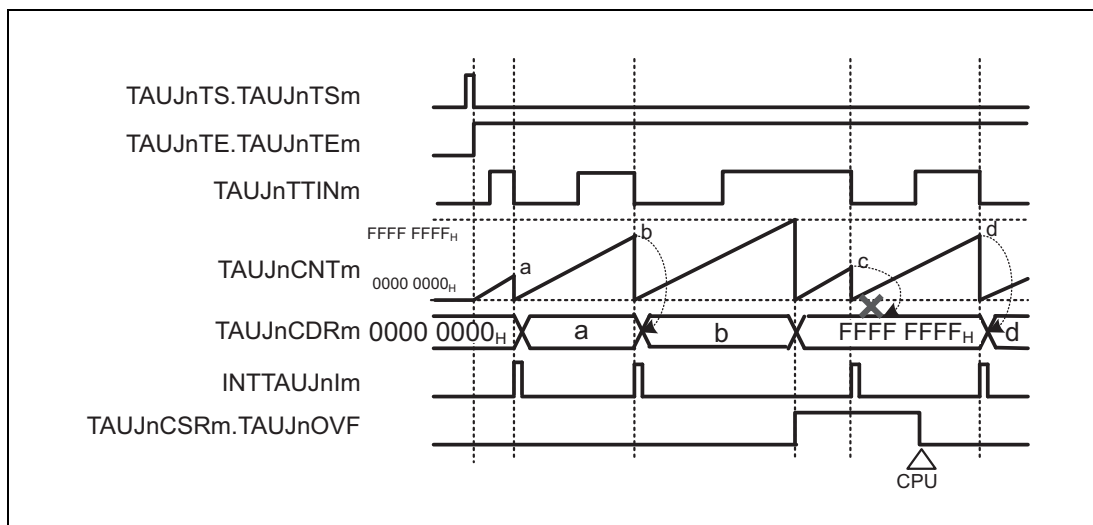
(4) $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 11_{\text{B}}$ 

Figure 24.28 $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 11_{\text{B}}$, $\text{TAUJnCMORm.TAUJnMD0} = 0$,
 $\text{TAUJnCMURm.TAUJnTIS}[1:0] = 00_{\text{B}}$

- When an overflow occurs, TAUJnCDRm is set to $\text{FFFF FFFF}_{\text{H}}$, and $\text{TAUJnCSRm.TAUJnOVF}$ is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and $\text{TAUJnCSRm.TAUJnOVF}$ remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.
- $\text{TAUJnCSRm.TAUJnOVF}$ is cleared by setting the $\text{TAUJnCSCm.TAUJnCLOV}$ bit to 1.

24.12.4 TAUJnTTINm Input Signal Width Measurement Function

24.12.4.1 Overview

Summary

This function measures the width of a TAUJnTTINm signal by starting counting on one edge of the TAUJnTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

- The operation mode must be set to capture and one-count mode, refer to **Table 24.50, Contents of the TAUJnCMORm register for TAUJnTTINm Input Signal Width Measurement Function**.
- TAUJnTTOUTm is not used for this function
- TAUJnCMORm.TAUJnMD should be set to 0.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. When a valid TAUJnTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm stop edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJnTTINm input start edge.

If the counter reaches FFFF FFFF_H before a valid TAUJnTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0]:

Table 24.49 Effects of an overflow

| TAUJnCMORm. TAUJnCOS[1:0] | When overflow occurs | | When a valid TAUJnTTINm input stop edge is detected | |
|------------------------------|----------------------------------|------------------------|---|------------------------|
| | TAUJnCDRm | TAUJnCSRm. TAUJnOVF | TAUJnCDRm and TAUJnCNTm | TAUJnCSRm. TAUJnOVF |
| 00 | Unchanged | 0 | TAUJnCNTm written to TAUJnCDRm | 1 |
| 01 | | 1 | | |
| 10 | Set to FFFF FFFF _H | 0 | TAUJnCNTm stops counting TAUJnCDRm unchanged | Unchanged |
| 11 | | 1 | | |

If an overflow is set (TAUJnCSRm.TAUJnOVF = 1) when TAUJnCMORm.TAUJnCOS[0] = 1, it can only be cleared by setting TAUJnCSCm.TAUJnCLOV = 1.

The combination of the value of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJnTTINm signal. However, if an overflow occurs multiple times before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

This function cannot be forcibly restarted.

This function is not supported by forced restarting

NOTE

When $\text{TAUJnCMORm.TAUJnCOS}[1] = 1$, the value of TAUJnCNTm is not written to TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

24.12.4.2 Equations

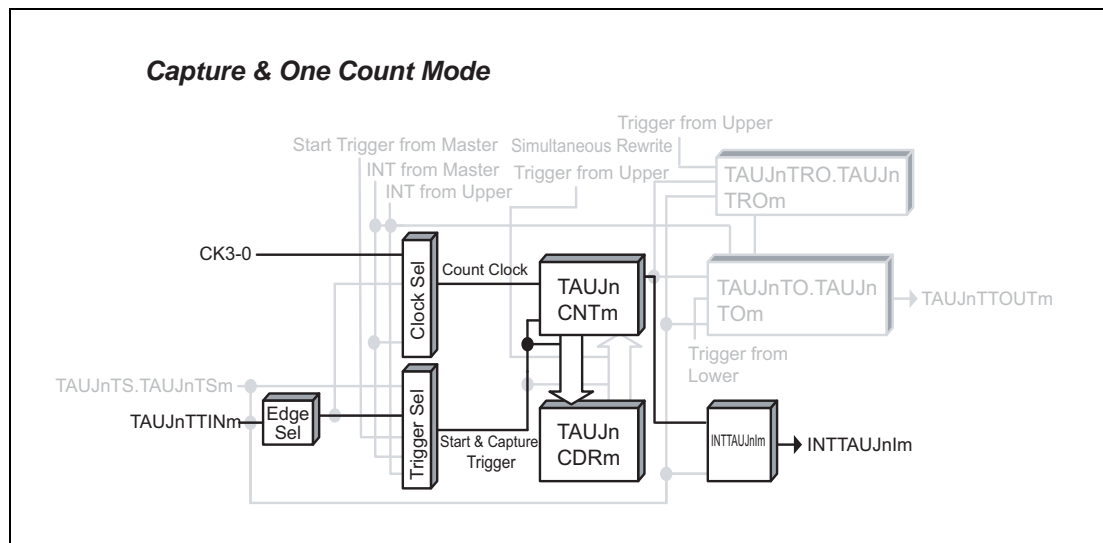
$$\text{TAUJnTTINm input signal width} = \text{count clock cycle} \times [(\text{TAUJnCSRm.TAUJnOVF} \times (\text{FFFF FFFF}_H + 1)) + \text{TAUJnCDRm capture value} + 1]$$
24.12.4.3 Block Diagram and General Timing Diagram

Figure 24.29 Block diagram for TAUJnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement ($\text{TAUJnCMURm.TAUJnTIS}[1:0] = 11_B$)
- When a valid TAUJnTTINm input is detected after an overflow, TAUJnCDRm is changed and $\text{TAUJnCSRm.TAUJnOVF}$ is set to 1 ($\text{TAUJnCMORm.TAUJnCOS}[1:0] = 00_B$)

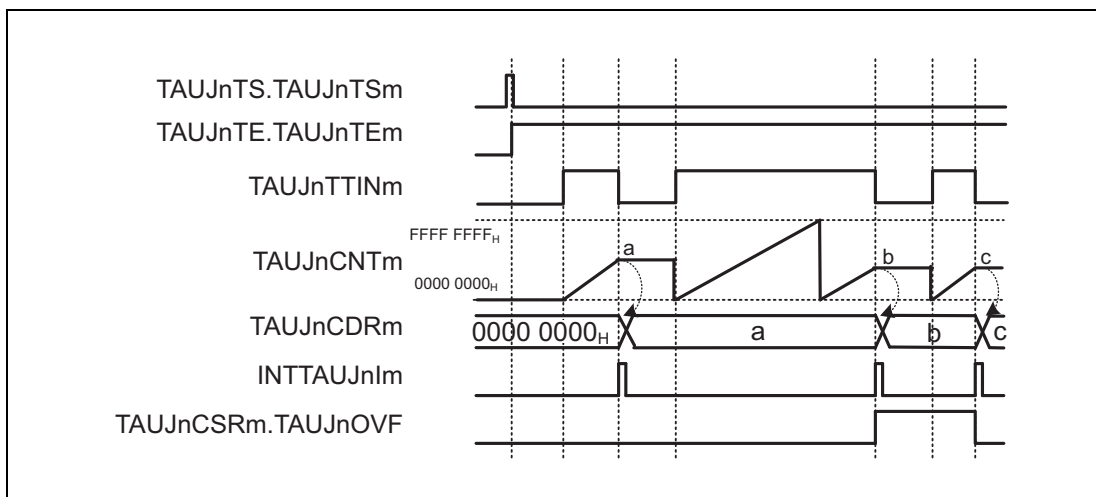


Figure 24.30 General Timing Diagram for TAUJnTTINm Input Signal Width Measurement Function

24.12.4.4 Register Settings

(1) TAUJnCMORM

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.50 Contents of the TAUJnCMORM register for TAUJnTTINm Input Signal Width Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 0 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 010 _B . |
| 7, 6 | TAUJnCOS[1:0] | See Table 24.49, Effects of an overflow. |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 0110 _B . |
| 0 | TAUJnMD0 | Write 0 _B . |

(2) TAUJnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.51 Contents of the TAUJnCMURm register for TAUJnTTINm Input Signal Width Measurement Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement) |

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Signal Width Measurement Function. Therefore, these registers must be set to 0.

Table 24.52 Simultaneous Rewrite Settings for TAUJnTTINm Input Signal Width Measurement Function

| Bit name | Setting |
|--------------------|--|
| TAUJnRDE.TAUJnRDEm | 0: Disables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0 |

24.12.4.5 Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function

Table 24.53 Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function

| | Operation | Status of TAUJn |
|-------------------------|--|---|
| Initial channel setting | Set the TAUJnCMORm register and TAUJnCMURm registers as described in Table 24.50, Contents of the TAUJnCMORm register for TAUJnTTINm Input Signal Width Measurement Function and Table 24.51, Contents of the TAUJnCMURm register for TAUJnTTINm Input Signal Width Measurement Function . The TAUJnCDRm register operates as a capture register. | Channel operation is stopped. |
| Restart | Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. When a TAUJnTTINm start edge is detected, TAUJnCNTm starts to count up. |
| During operation | The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.TAUJnCLOV bit can be set to 1. | TAUJnCNTm starts to count up from 0000 0000 _H . When a TAUJnTTINm valid edge is detected: <ul style="list-style-type: none"> • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and retains its value • INTTAUJnIm is then generated. • Counting stops at the “value transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. Afterwards, this procedure is repeated. |
| Stop operation | Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values. |

24.12.4.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

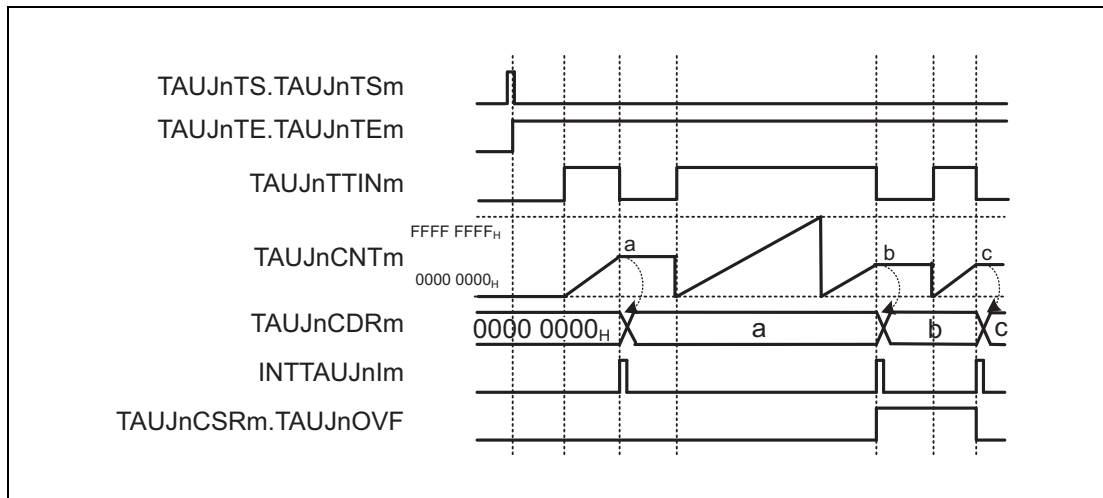


Figure 24.31 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF remains = 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is written to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

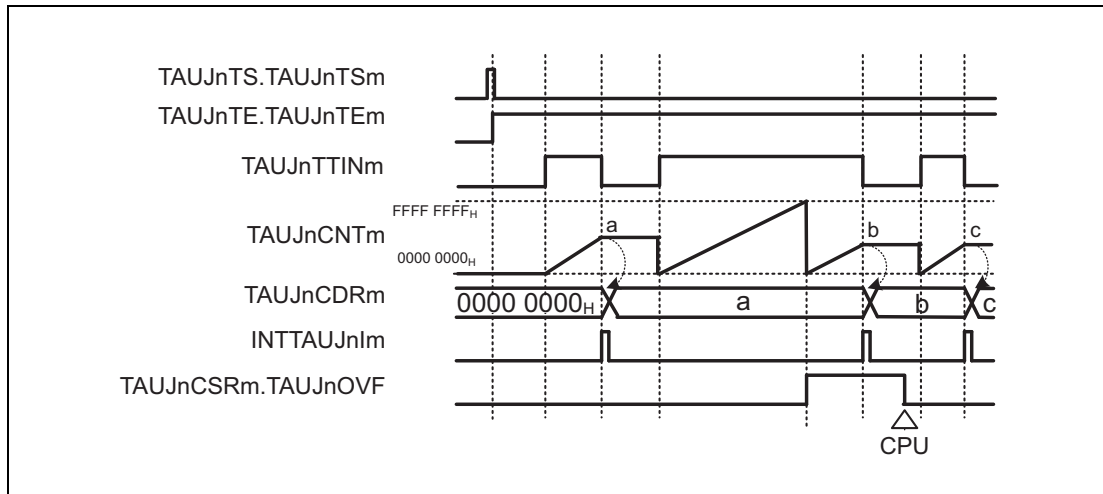
(2) $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 01_{\text{B}}$ 

Figure 24.32 $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 01_{\text{B}}$, $\text{TAUJnCMORm.TAUJnMD0} = 0$,
 $\text{TAUJnCMURm.TAUJnTIS}[1:0] = 11_{\text{B}}$

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and $\text{TAUJnCSRm.TAUJnOVF}$ is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is written to TAUJnCDRm .
- $\text{TAUJnCSRm.TAUJnOVF}$ is only cleared by a CPU command (by setting $\text{TAUJnCSCm.TAUJnCLOV}$ bit = 1).

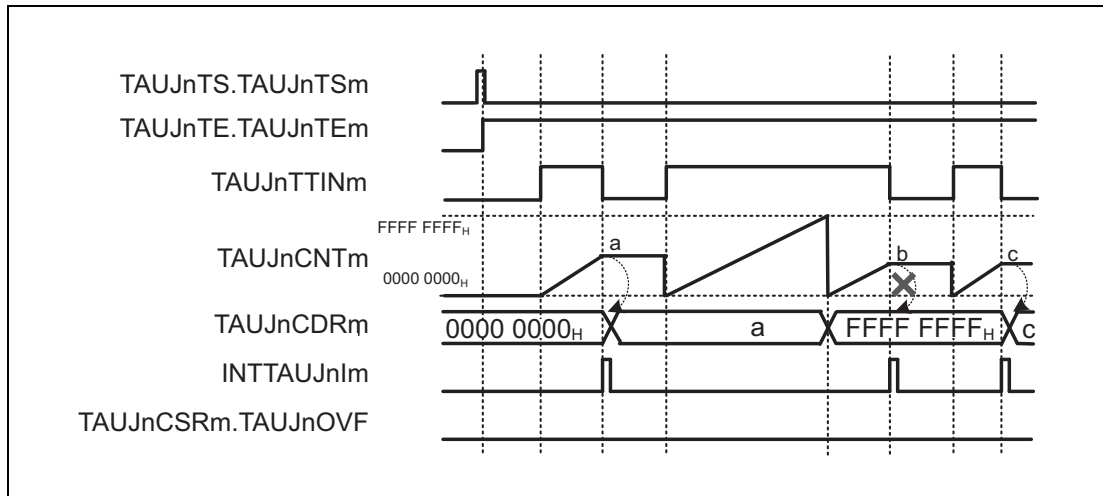
(3) $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 10_{\text{B}}$ 

Figure 24.33 $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 10_{\text{B}}$, $\text{TAUJnCMORm.TAUJnMD0} = 0$,
 $\text{TAUJnCMURm.TAUJnTIS}[1:0] = 11_{\text{B}}$

- When an overflow occurs, TAUJnCDRm is set to $\text{FFFF FFFF}_{\text{H}}$ and $\text{TAUJnCSRm.TAUJnOVF}$ remains = 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and $\text{TAUJnCSRm.TAUJnOVF}$ remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.

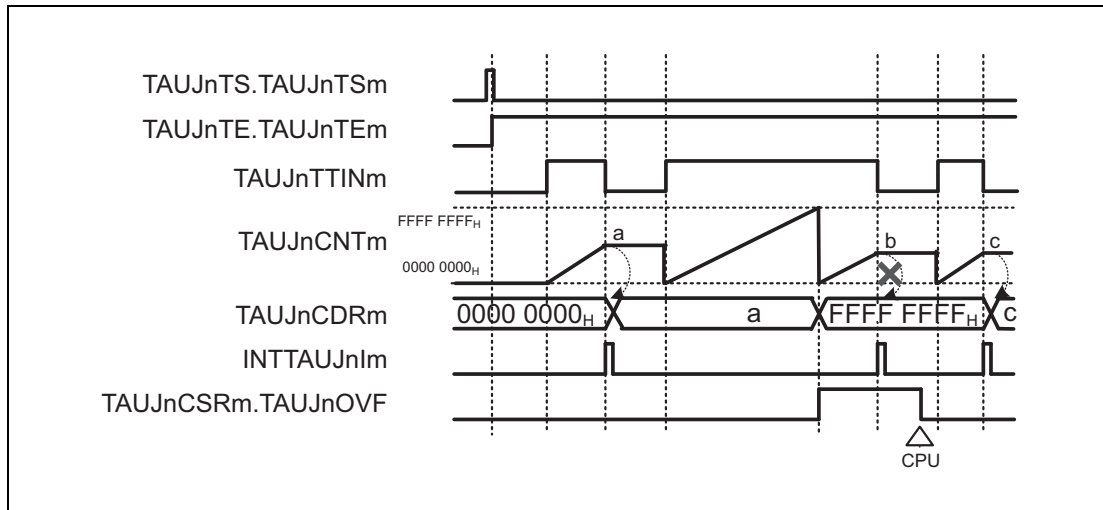
(4) $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 11_{\text{B}}$ 

Figure 24.34 $\text{TAUJnCMORm.TAUJnCOS}[1:0] = 11_{\text{B}}$, $\text{TAUJnCMORm.TAUJnMD0} = 0$,
 $\text{TAUJnCMURm.TAUJnTIS}[1:0] = 11_{\text{B}}$

- When an overflow occurs, TAUJnCDRm is set to $\text{FFFF FFFF}_{\text{H}}$, and $\text{TAUJnCSRm.TAUJnOVF}$ is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and $\text{TAUJnCSRm.TAUJnOVF}$ remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.
- $\text{TAUJnCSRm.TAUJnOVF}$ is cleared by setting the $\text{TAUJnCSCm.TAUJnCLOV}$ bit to 1.

24.12.5 TAUJnTTINm Input Position Detection Function

24.12.5.1 Overview

Summary

This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUJnTTINm signal.

Prerequisites

- The operation mode must be set to count capture mode, refer to **Table 24.54, Contents of the TAUJnCMORm register for TAUJnTTINm Input Position Detection Function**.
- TAUJnTTOUtm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter starts to count from 0000 0000_H. When a valid TAUJnTTINm input edge is detected, the current TAUJnCNTm value is written to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter continues to count.

When the counter reaches FFFF FFFF_H, it restarts from 0000 0000_H.

NOTE

The TAUJnTTINm input signal is sampled at the frequency of the operation clock specified by the TAUJnCMORm.TAUJnCKS[1:0] bits. As a result, the output cycle of the TAUJnTTOUtm has an error of ±1 operation clock cycle.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For the detail, see **Section 24.9, TAUJnTTOUtm Output and INTTAUJnIm Generation when Counter Starts or Restarts**.

24.12.5.2 Equations

Function duration at a TAUJnTTINm input pulse =
count clock cycle × (TAUJnCDRm capture value + 1)

24.12.5.3 Block Diagram and General Timing Diagram

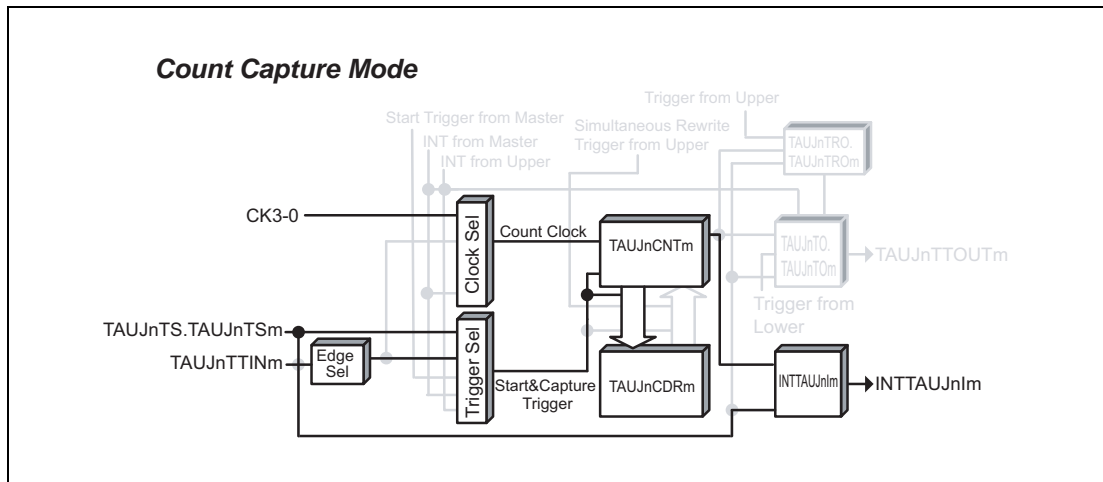


Figure 24.35 Block Diagram of TAUJnTTINm Input Position Detection Function

The following settings apply to the general timing diagram:

- INTTAUJnIm not generated at operation start (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

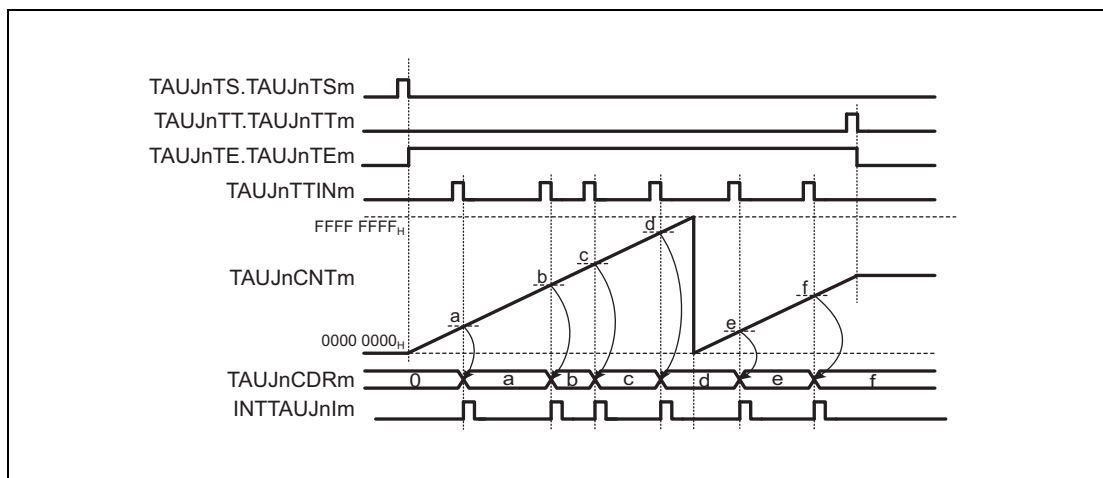


Figure 24.36 General Timing Diagram for TAUJnTTINm Input Position Detection Function

24.12.5.4 Register Settings

(1) TAUJnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.54 Contents of the TAUJnCMORm register for TAUJnTTINm Input Position Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 0 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 001 _B . |
| 7, 6 | TAUJnCOS[1:0] | Write 01 _B . |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 1011 _B . |
| 0 | TAUJnMD0 | 0: INTTAUJnIm not generated at operation start 1: Generates INTTAUJnIm at operation start |

(2) TAUJnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.55 Contents of the TAUJnCMURm register for TAUJnTTINm Input Position Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection |

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Position Detection Function. Therefore, these registers must be set to 0.

Table 24.56 Simultaneous Rewrite Settings for TAUJnTTINm Input Period Count Detection Function

| Bit name | Setting |
|--------------------|--|
| TAUJnRDE.TAUJnRDEm | 0: Disables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0 |

24.12.5.5 Operating Procedure for TAUJnTTINm Input Position Detection Function

Table 24.57 Operating Procedure for TAUJnTTINm Input Position Detection Function

| | Operation | Status of TAUJn |
|-------------------------|--|--|
| Initial channel setting | Set the TAUJnCMORm register and TAUJnCMURm registers as described in Table 24.54, Contents of the TAUJnCMORm register for TAUJnTTINm Input Position Detection Function and Table 24.55, Contents of the TAUJnCMURm register for TAUJnTTINm Input Position Detection Function . The TAUJnCDRm register operates as a capture register. | Channel operation is stopped. |
| Restart | Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is set to 1 and the counter starts. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1. |
| During operation | The TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. | TAUJnCNTm starts to count up from 0000 0000 _H . When a TAUJnTTINm valid edge is detected: <ul style="list-style-type: none"> • TAUJnCNTm transfers (captures) its value to TAUJnCDRm • INTTAUJnIm is output. • The counter value is not cleared to 0000 0000_H and TAUJnCNTm continues count operation. Afterwards, this procedure is repeated. If the TAUJnCNTm reaches FFFF FFFF _H , it restarts counting from 0000 0000 _H . |
| Stop operation | Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value. |

24.12.5.6 Specific Timing Diagrams

(1) Operation stop and restart

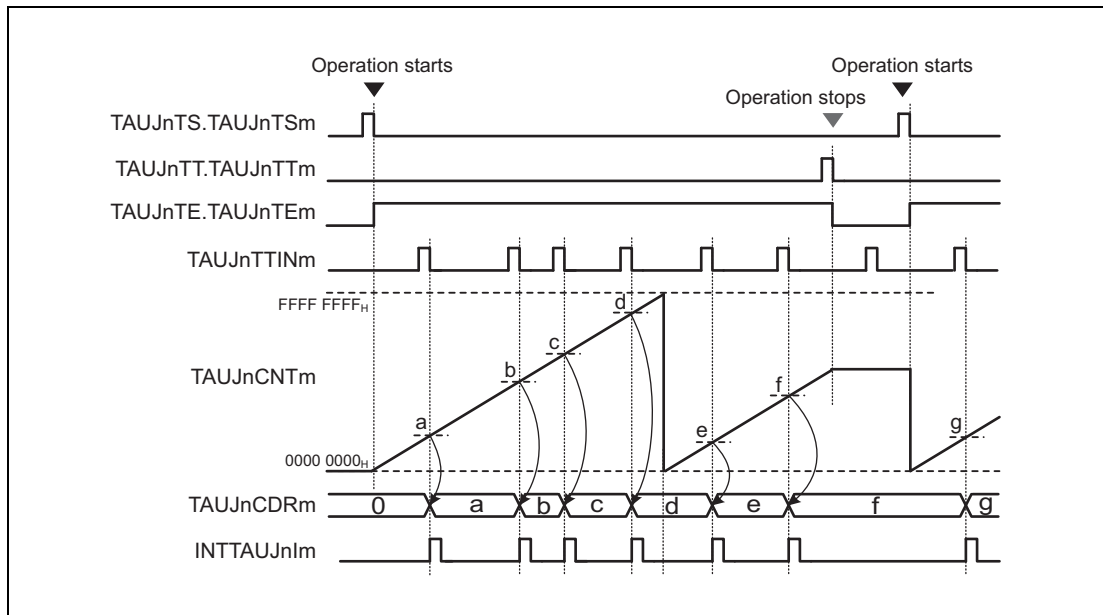


Figure 24.37 Operation Stop and Restart, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- The counter can be stopped by setting TAUJnTT.TAUJnTTM to 1, which in turn sets TAUJnTE.TAUJnTEM to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSM to 1. TAUJnCNTm restarts to count from 0000 0000_H.

24.12.6 TAUJnTTINm Input Period Count Detection Function

24.12.6.1 Overview

Summary

This function measures the cumulative width of a TAUJnTTINm input signal.

Prerequisites

- The operation mode must be set to capture and gate count mode, refer to **Table 24.58, Contents of the TAUJnCMORm Register for TAUJnTTINm Input Period Count Detection Function**.
- TAUJnTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJnTTINm input edge.

When a valid TAUJnTTINm input start edge is detected, the counter starts to count from 0000 0000_H.

When a valid TAUJnTTINm input stop edge is detected, the current TAUJnCNTm value is written to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJnTTINm input start edge is detected.

When the next valid TAUJnTTINm input start edge is detected, the counter restarts counting from the stop value.

If the counter reaches FFFF FFFF_H the counter restarts from 0000 0000_H.

In this function, a forced restart cannot be performed.

NOTE

The TAUJnTTINm input signal is sampled at the frequency of the operating clock, specified by the TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits:

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJnTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TIS[1:0] = 11_B, the TAUJnTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

24.12.6.2 Equations

Cumulative TAUJnTTINm input width =
count clock cycle × (TAUJnCDRm capture value + 1)

24.12.6.3 Block Diagram and General Timing Diagram

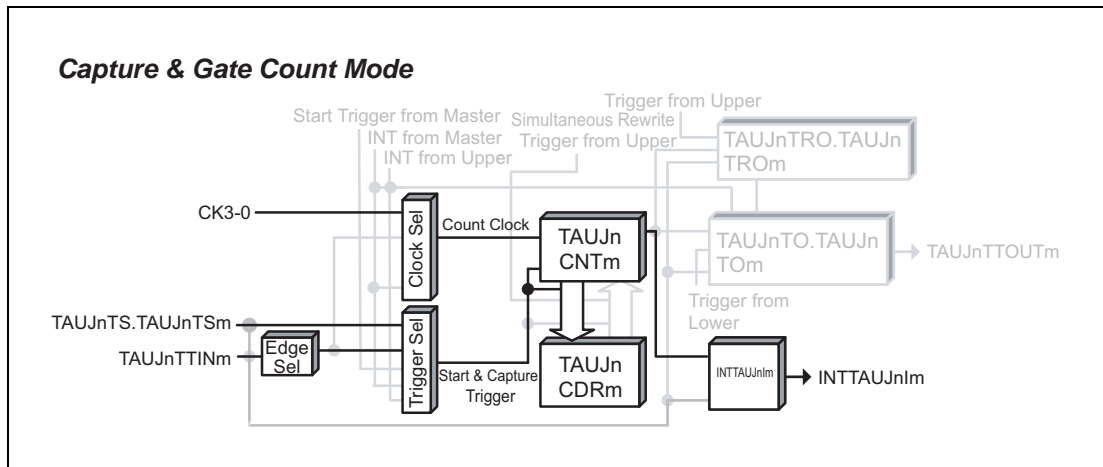


Figure 24.38 Block diagram for TAUJnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement
(TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

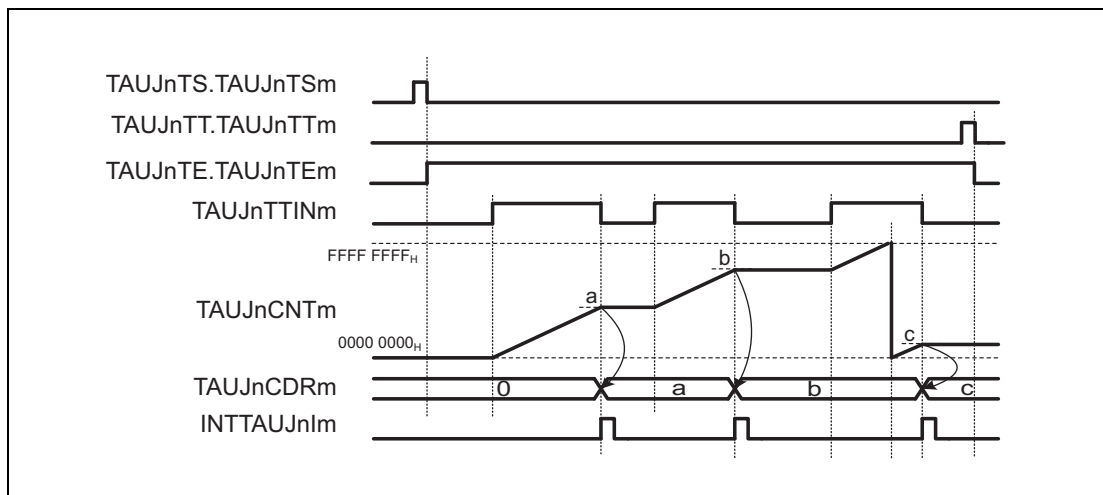


Figure 24.39 General Timing Diagram for TAUJnTTINm Input Period Count Detection Function

24.12.6.4 Register Settings

(1) TAUJnCMORm

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|-----|---------------|-----|---|--------------|-----|-----|-----|----------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.58 Contents of the TAUJnCMORm Register for TAUJnTTINm Input Period Count Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 0 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 010 _B . |
| 7, 6 | TAUJnCOS[1:0] | Write 01 _B . |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 1101 _B . |
| 0 | TAUJnMD0 | Write 0 _B . |

(2) TAUJnCMURm

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.59 Contents of the TAUJnCMURm Register for TAUJnTTINm Input Period Count Detection Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement) |

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Period Count Detection Function. Therefore, these registers must be set to 0.

Table 24.60 Simultaneous Rewrite Settings for TAUJnTTINm Input Period Count Detection Function

| Bit name | Setting |
|--------------------|--|
| TAUJnRDE.TAUJnRDEm | 0: Disables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0 |

24.12.6.5 Operating Procedure for TAUJnTTINm Input Period Count Detection Function

Table 24.61 Operating Procedure for TAUJnTTINm Input Period Count Detection Function

| Operation | Status of TAUJn |
|---|--|
| Initial channel setting Set the TAUJnCMORm register and TAUJnCMURm registers as described in Table 24.58, Contents of the TAUJnCMORm Register for TAUJnTTINm Input Period Count Detection Function and Table 24.59, Contents of the TAUJnCMURm Register for TAUJnTTINm Input Period Count Detection Function . The TAUJnCDRm register operates as a capture register. | Channel operation is stopped. |
| Restart Start operation Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. |
| During operation TAUJnTTINm edge detection The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. | When a TAUJnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts to count up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUJnCDRm and INTTAUJnIm is generated. Counting stops at the "value transferred to TAUJnCDRm + 1" and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. If the TAUJnCNTm reaches FFFF FFFF _H , it restarts counting from 0000 0000 _H . Afterwards, this procedure is repeated. |
| Stop operation Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value. |

24.12.6.6 Specific Timing Diagrams

(1) Operation stop and restart

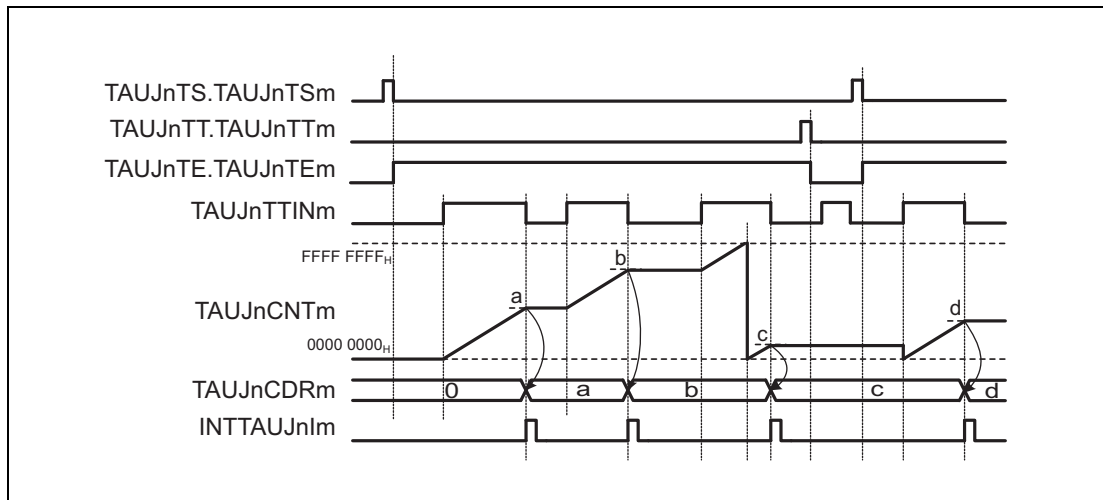


Figure 24.40 Operation Stop and Restart, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSM to 1. TAUJnCNTm restarts to count from 0000 0000_H.

24.13 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the TAUJ. For a general overview of synchronous channel operation, see **Section 24.2, Overview**.

24.13.1 PWM output function

24.13.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJnTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode, refer to **Table 24.62, Contents of the TAUJnCMORm register for the Master Channel of the PWM Output Function**.
- The operating mode for the slave channels should be set to one count mode, refer to **Table 24.65, Contents of the TAUJnCMORm register for the Slave Channel of the PWM Output Function**.
- TAUJnTTOUTm is not used for this function
- The channel output mode for the slave channels should be set to Synchronous Channel Output Mode 1, refer to **Section 24.7, Channel Output Modes**.

Description

The counters are enabled by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is written to TAUJnCNTm and the counters start to count down from these values. INTTAUJnIm is generated on the master channel and TAUJnTTOUTm (slave) is set or reset to realize the PWM output.

- Master channel:

When the counter of the master channel reaches 0000 0000_H, pulse cycle time has elapsed and INTTAUJnIm is generated. The TAUJnCDRm value is written to TAUJnCNTm, and the counter counts down.

- Slave channel(s):

INTTAUJnIm generated on the master channel triggers the counter of the slave channel(s). The current value of TAUJnCDRm (slave) is written to TAUJnCNTm (slave) and the counter starts to count down from this value. The TAUJnTTOUTm signal is set to the active level.

When the counter reaches 0000 0000_H, i.e. duty time has elapsed, INTTAUJnIm is generated and the TAUJnTTOUTm signal is set to the inactive level. The counter returns to FFFF FFFF_H and awaits the next INTTAUJnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channel(s), which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJnTTOUTm of master and slave

channel(s) stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSM to 1.

Conditions

Simultaneous rewrite can be used with this function. Please refer to **Section 24.6, Simultaneous Rewrite**.

24.13.1.2 Equations

$$\text{Pulse cycle} = (\text{TAUJnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$\text{Duty cycle [\%]} = (\text{TAUJnCDRm (slave)} / (\text{TAUJnCDRm (master)} + 1)) \times 100$$

- Duty cycle = 0%
 $\text{TAUJnCDRm (slave)} = 0000\ 0000_{\text{H}}$
- Duty cycle = 100%
 $\text{TAUJnCDRm (slave)} \geq \text{TAUJnCDRm (master)} + 1$

24.13.1.3 Block Diagram and General Timing Diagram

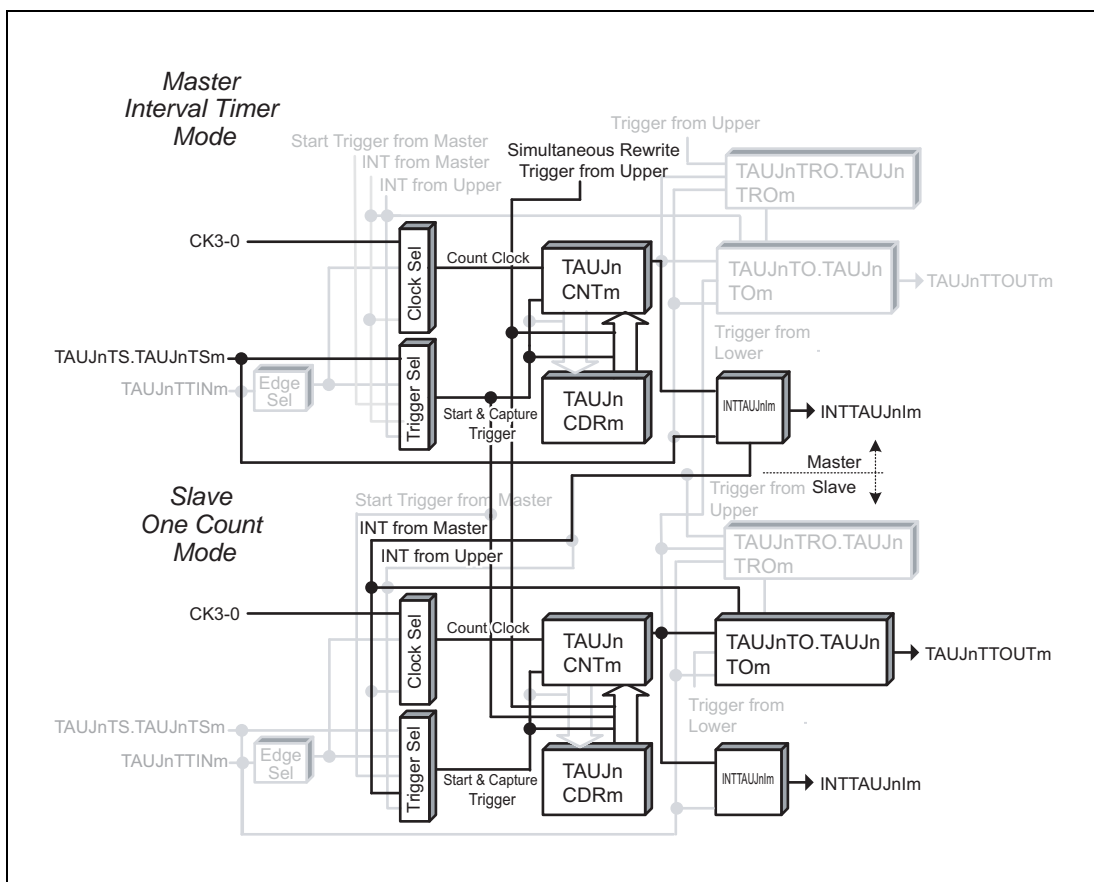


Figure 24.41 Block diagram for PWM Output Function

The following settings apply to the general timing diagram:

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

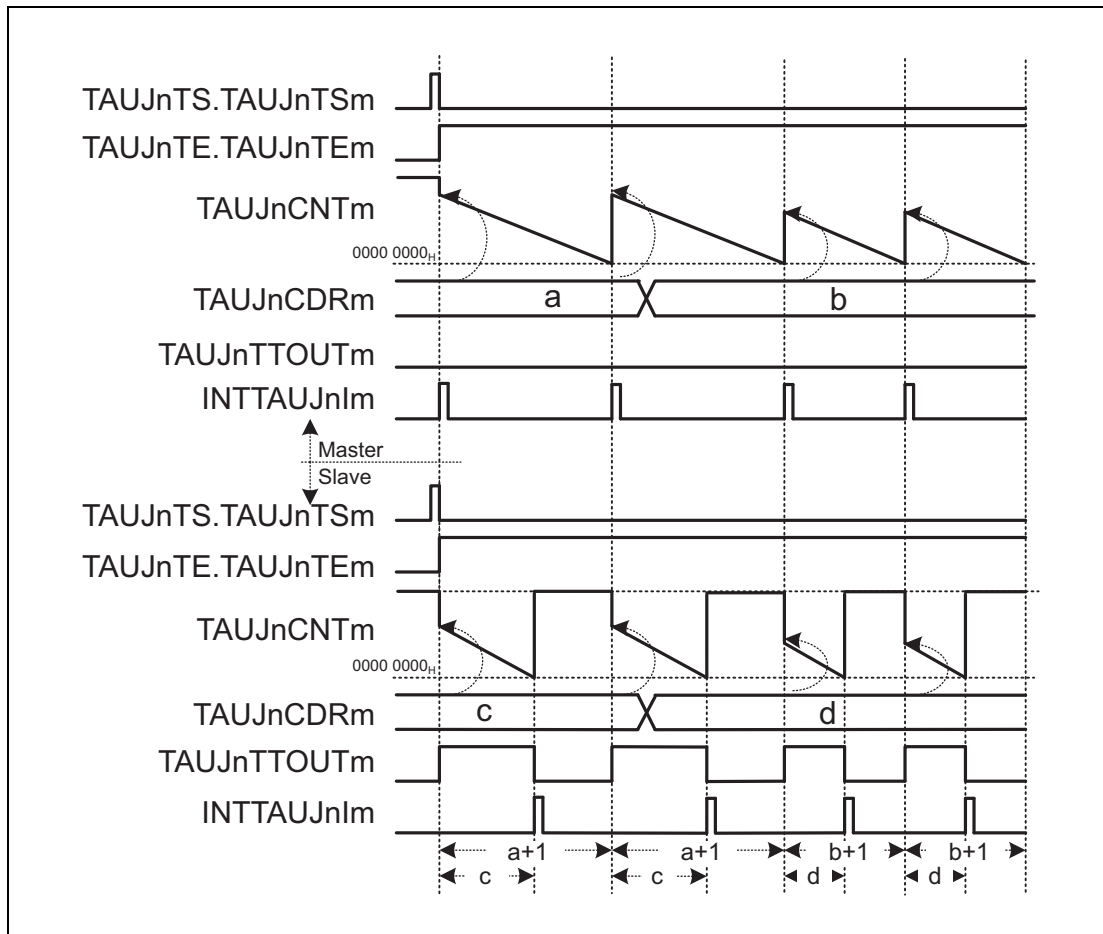


Figure 24.42 General Timing Diagram for PWM Output Function

NOTE

- The interval between the channel starting to count and an interrupt being generated is the value of corresponding TAUJnCDRm + 1.
- The TAUJnTTOUTm signal from slave channel rises 1 cycle of the counter clock after the rising edge of INTTAUJnIm from the master channel.

24.13.1.4 Register Settings for the Master Channel

(1) TAUJnCMORm for the master channel

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.62 Contents of the TAUJnCMORm register for the Master Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bit of the master and slave channel(s) must be identical. |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 1 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 000 _B . |
| 7, 6 | TAUJnCOS[1:0] | Write 00 _B . |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 0000 _B . |
| 0 | TAUJnMD0 | Write 1 _B . |

(2) TAUJnCMURm for the master channel

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.63 Contents of the TAUJnCMURm register for the Master Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 00: Not used so set to 00 |

(3) Channel output mode for the master channel

The channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 24.64 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

| Bit name | Setting |
|--------------------|---|
| TAUJnRDE.TAUJnRDEm | 1: Enables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: The simultaneous rewrite trigger signal is generated when the master channel starts counting |

24.13.1.5 Register Settings for the Slave Channel(s)

(1) TAUJnCMORm for the slave channel(s)

| | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----|---------------|-----|----------|---------------|-----|---------------|-----|-----|--------------|-----|-----|-----|----------|-----|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TAUJnCKS[1:0] | | TAUJnCCS[1:0] | | TAUJnMAS | TAUJnSTS[2:0] | | TAUJnCOS[1:0] | | — | TAUJnMD[4:1] | | | | TAUJnMD0 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |

Table 24.65 Contents of the TAUJnCMORm register for the Slave Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 15, 14 | TAUJnCKS[1:0] | Selects an operating clock. 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bit of the master and slave channel(s) must be identical. |
| 13, 12 | TAUJnCCS[1:0] | Write 00 _B . |
| 11 | TAUJnMAS | Write 0 _B . |
| 10 to 8 | TAUJnSTS[2:0] | Write 100 _B . |
| 7, 6 | TAUJnCOS[1:0] | Write 00 _B . |
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 1 | TAUJnMD[4:1] | Write 0100 _B . |
| 0 | TAUJnMD0 | Write 1 _B . |

(2) TAUJnCMURm for the slave channel(s)

| | | | | | | | | |
|-------------------|-----|---|---|---|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TAUJnTIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R | R | R | R | R | R/W | R/W |

Table 24.66 Contents of the TAUJnCMURm register for the Slave Channel of the PWM Output Function

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TAUJnTIS[1:0] | 00: Not used so set to 00 |

(3) Channel output mode for the slave channel(s)**Table 24.67 Control Bit Settings for Synchronous Channel Output Mode 1**

| Bit Name | Setting |
|--------------------|--|
| TAUJnTOE.TAUJnTOEm | Write 1 _B . |
| TAUJnTOM.TAUJnTOMm | Write 1 _B . |
| TAUJnTOC.TAUJnTOCm | Write 0 _B . |
| TAUJnTOL.TAUJnTOLm | 0: Positive logic 1: Negative logic |

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 24.68 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

| Bit Name | Setting |
|--------------------|---|
| TAUJnRDE.TAUJnRDEm | 1: Enables simultaneous rewrite |
| TAUJnRDM.TAUJnRDMm | 0: The simultaneous rewrite trigger signal is generated when the master channel starts counting |

24.13.1.6 Operating Procedure for PWM Output Function

Table 24.69 Operating Procedure for PWM Output Function

| | Operation | Status of TAUJn |
|-------------------------|---|--|
| Initial channel setting | Master channel: set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in Section 24.13.1.4, Register Settings for the Master Channel. | Channel operation is stopped. |
| | Slave channel: set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in Section 24.13.1.5, Register Settings for the Slave Channel(s). | |
| | Set the values of the TAUJnCDRm registers of all channels | |
| Restart | Start operation Set TAUJnTS.TAUJnTSm of the master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUJnIm is generated on the master channel and TAUJnTTOUTm (slave) is set. |
| | During operation TAUJnCDRm can be changed at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm can be read at any time. TAUJnRDT.TAUJnRDTm can be changed during operation. | TAUJnCNTm of the master channel loads TAUJnCDRm and counts down. When the counter reaches 0000 0000 _H : <ul style="list-style-type: none"> • INTTAUJnIm (master) is generated • TAUJnCNTm (master) loads the TAUJnCDRm value and continues count operation • TAUJnCNTm (slave) loads the TAUJnCDRm value and counts down • TAUJnTTOUTm (slave) is set to the active level. When TAUJnCNTm (slave) reaches 0000 0000 _H : <ul style="list-style-type: none"> • INTTAUJnIm (slave) is generated • TAUJnTTOUTm (slave) is set to the inactive level. |
| Stop operation | Set TAUJnTT.TAUJnTTm of the master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0. | TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values. |

24.13.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

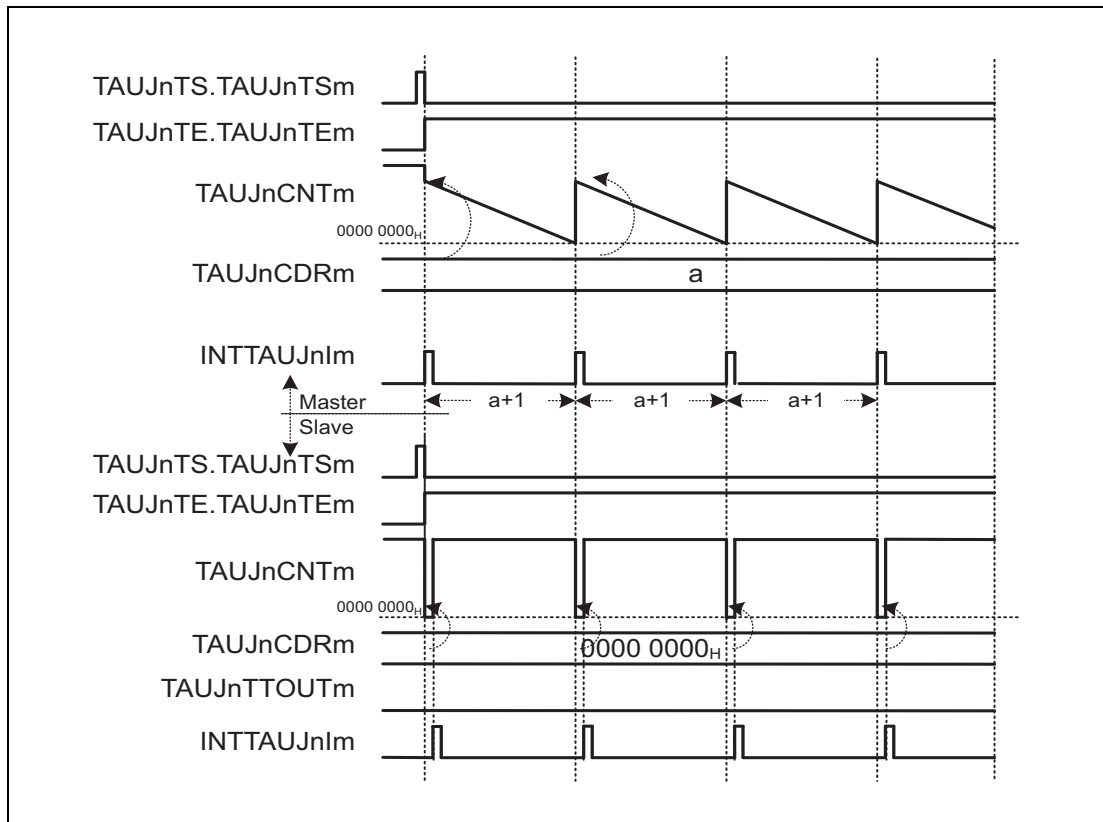


Figure 24.43 TAUJnCDRm (Slave) = 0000 0000_H, Positive Logic
(TAUJnTOL.TAUJnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUJnIm), 0000 0000_H is written to TAUJnCNTm (slave). Therefore, an interrupt (INTTAUJnIm) is simultaneously generated at the slave channel and TAUJnTTOUTm remains at not active state.
- The value of TAUJnCDRm is loaded into TAUJnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

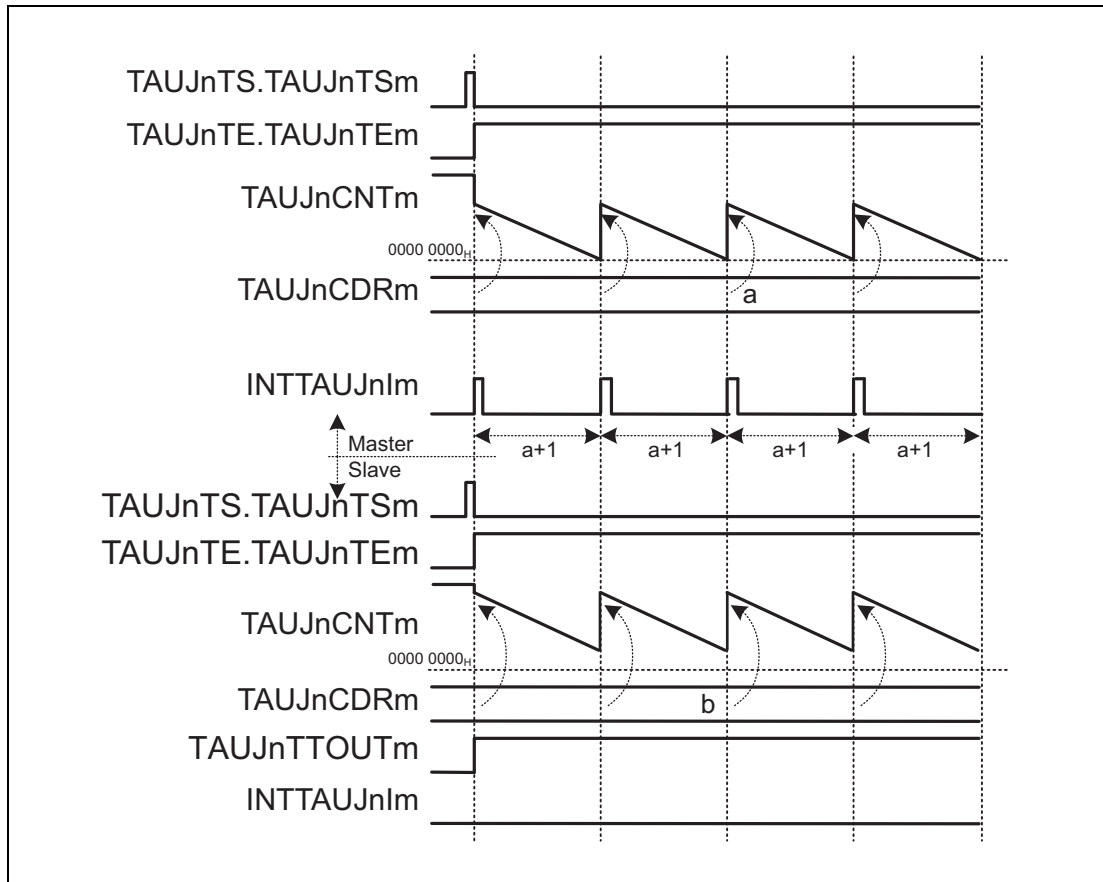


Figure 24.44 TAUJnCDRm (Slave) ≥ TAUJnCDRm (Master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

If the TAUJnCDRm (slave) value is greater than the TAUJnCDRm (master) value, no interrupt occurs because the counter of the slave channel does not reach 0000 0000_H. TAUJnTTOUTm remains active.

(3) Operation stop and restart

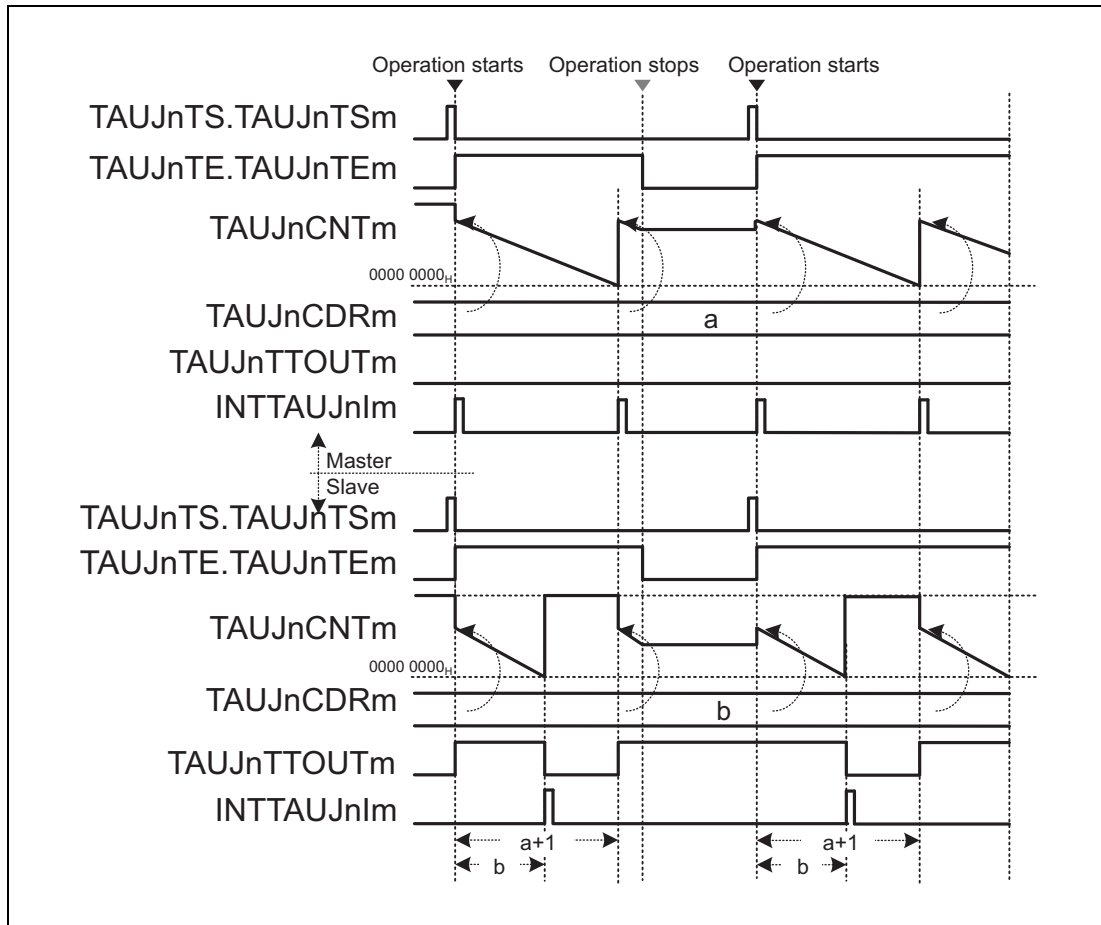


Figure 24.45 Operation Stop and Restart, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of master and slave channels to 1. This sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm of every channel stop and retain their current values. No interrupt occurs.
- The counter can be restarted by setting TAUJnTS.TAUJnTSM of master and slave channels to 1. TAUJnCDRm value of master and slave channels is loaded into TAUJnCNTm. The counter starts to count down from this value.

Section 25 Motor Control Timer (TSG3)

The Motor Control Timer (TSG3) is 18-bit counter which can generate the output corresponding to various motor control systems. TSG3 is implemented 2 units.

25.1 Features of RH850/P1M-E TSG3

25.1.1 Number of Units

This microcontroller has the following number of units of TSG3 units.

Table 25.1 Number of Units

| Product | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 2 | 2 |
| Name | TSG3n (n = 0, 1) | TSG3n (n = 0, 1) |

Table 25.2 Index

| Index | Description |
|-------|---|
| n | Throughout this section, the individual TSG3 units are identified by the index “n” (n = 0, 1), for example, TSG3nCTL0 refers to the TSG3n control register 0. |
| m, k | Throughout this section, the variables used for description are indicated by the index “m” or “k”, for example, TSG3nCMPmE refers to a specific compare register. |

25.1.2 Register Base Address

TSG3 base addresses are listed in the following table.

TSG3 register addresses are given as offsets from the base addresses.

Table 25.3 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <TSG30_base> | FFE7 0000 _H |
| <TSG31_base> | FFE7 1000 _H |

25.1.3 Clock Supply

Clock supply by and to TSG3 is listed in the following table.

Table 25.4 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|-------------------------------------|
| TSG3n | PCLK | High-speed peripheral clock CLK_HSB |

25.1.4 Interrupt Requests

TSG3 interrupt requests are listed in the following table.

Table 25.5 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Number | DMA Trigger Number | DTS Trigger Number |
|---------------------|--|------------------|--------------------|--------------------|
| TSG30 | | | | |
| INTTSG30I0 | TSG30 compare match interrupt 0/ TSG30 period interrupt (in HT-PWM mode only) | 40 | — | — |
| INTTSG30I1 | TSG30 compare match interrupt 1 | 41 | — | — |
| INTTSG30I2 | TSG30 compare match interrupt 2 | 42 | — | — |
| INTTSG30I3 | TSG30 compare match interrupt 3 | 43 | — | — |
| INTTSG30I4 | TSG30 compare match interrupt 4 | 44 | — | — |
| INTTSG30I5 | TSG30 compare match interrupt 5 | 45 | — | — |
| INTTSG30I6 | TSG30 compare match interrupt 6 | 46 | — | — |
| INTTSG30I7 | TSG30 compare match interrupt 7 | 47 | — | — |
| INTTSG30I8 | TSG30 compare match interrupt 8 | 48 | — | — |
| INTTSG30I9 | TSG30 compare match interrupt 9 | 49 | — | — |
| INTTSG30I10 | TSG30 compare match interrupt 10 | 50 | — | — |
| INTTSG30I11 | TSG30 compare match interrupt 11 | 51 | 99 | 99 |
| INTTSG30I12 | TSG30 compare match interrupt 12 | 52 | 100 | 100 |
| INTTSG30IPEK | TSG30 peak interrupt | 53 | 101 | 101 |
| INTTSG30IVLY | TSG30 valley interrupt | 54 | 102 | 102 |
| INTTSG30IER | TSG30 error interrupt | 55 | — | — |
| INTTSG30IWN | TSG30 warning interrupt | 56 | — | — |
| TSG31 | | | | |
| INTTSG31I0 | TSG31 compare match interrupt 0/ TSG31 period interrupt (in HT-PWM mode only) | 57 | — | — |
| INTTSG31I1 | TSG31 compare match interrupt 1 | 58 | — | — |
| INTTSG31I2 | TSG31 compare match interrupt 2 | 59 | — | — |
| INTTSG31I3 | TSG31 compare match interrupt 3 | 60 | — | — |
| INTTSG31I4 | TSG31 compare match interrupt 4 | 61 | — | — |
| INTTSG31I5 | TSG31 compare match interrupt 5 | 62 | — | — |
| INTTSG31I6 | TSG31 compare match interrupt 6 | 63 | — | — |
| INTTSG31I7 | TSG31 compare match interrupt 7 | 64 | — | — |
| INTTSG31I8 | TSG31 compare match interrupt 8 | 65 | — | — |
| INTTSG31I9 | TSG31 compare match interrupt 9 | 66 | — | — |
| INTTSG31I10 | TSG31 compare match interrupt 10 | 67 | — | — |
| INTTSG31I11 | TSG31 compare match interrupt 11 | 68 | 103 | 103 |
| INTTSG31I12 | TSG31 compare match interrupt 12 | 69 | 104 | 104 |
| INTTSG31IPEK | TSG31 peak interrupt | 70 | 105 | 105 |
| INTTSG31IVLY | TSG31 valley interrupt | 71 | 106 | 106 |
| INTTSG31IER | TSG31 error interrupt | 72 | — | — |
| INTTSG31IWN | TSG31 warning interrupt | 73 | — | — |

25.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

25.1.6 External Input/Output Signals

External input/output signals of TSG3 are listed in the following table.

Table 25.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|--------------------------|-----|------------------------|---|
| TSG30 | | | |
| TSG30CLKI | I | Clock enable input | TSG30CLKI |
| TSG30PTSI0 to TSG30PTSI2 | I | External pattern input | ENCA0E0, ENCA0E1, ENCA0EC ^{*1} |
| TSG30O0 to TSG30O7 | O | Timer output | TSG30O0 to TSG30O7 |
| TSG31 | | | |
| TSG31CLKI | I | Clock enable input | TSG31CLKI |
| TSG31PTSI0 to TSG31PTSI2 | I | External pattern input | ENCA1E0, ENCA1E1, ENCA1EC ^{*1} |
| TSG31O0 to TSG31O7 | O | Timer output | TSG31O0 to TSG31O7 |

Note 1. External pattern input is shared with ENCA_n input.

25.2 Overview

25.2.1 Functional Overview

The TSG3n is an 18-bit timer counter with various motor control functions.

- Count clock resolution: Minimum 12.5 ns (count clock = 80 MHz)
- Operating mode corresponding to various motor control methods
- Compare registers with reload buffer
- 10-bit dead time counter
 - Dead time counter with reload buffer
 - Independent dead time can be set for positive to negative phase change and negative to positive phase change.
- A/D conversion trigger signal generation
 - Three A/D conversion trigger signals can be generated by the compare registers TSG3nDCMP0E, TSG3nDCMP1E, and TSG3nDCMP2E.
 - Skipping function of A/D conversion trigger signals TSG3nADTRG0 and TSG3nADTRG1 can be set independently. The skipping ratio can be selected among 1/1, 1/2, 1/4, and 1/8.
 - The dedicated pin (TSG3nO7) can be used to output the toggle or diagnostic signal set by the TSG3nADTRG0 signal and reset by the TSG3nADTRG1 signal.
- Interrupt skipping
 - Skipping rate: 1/1 to 1/32
- Forced output stop function
 - Using the timer option (TAPA) function allows high-impedance control of the TSG3nO1 to TSG3nO6 pin outputs.
- Compare value setting
 - Reload (simultaneous rewrite) or anytime rewrite can be selected.
- Reload mode
 - Writing to TSG3nCMP1E (setting the reload request flag (TSG3nRSF)) enables reload, and allows simultaneous transfer of values for multiple registers.
 - Data can be transferred at peak/valley/peak or valley reload timing
 - Reload request flag (TSG3nRSF)
 - Register address assignment allowing DMA transfer
 - Reload skipping
- HT-PWM mode
 - 0 to 100% PWM duty cycle output is possible (with possible dead time reduction).
 - The LSB in the compare register can be used to append an additional pulse to the PWM output during count up, thus improving the output resolution without extra load on software.

- 120-DC control
 - Semi-automatic drive function (trigger signal can be generated by an offset in conjunction with two-phase encoder, three-phase encoder, or ENCA).
- Three-phase encoder function (hall sensor signals can be input).
- Active level of the output pins TSG3nO1 to TSG3nO6 can be set individually.
- Fail-safe function (warning interrupt or error interrupt can be generated)
 - Simultaneous active output detect function for positive and inverse phases.
 - Abnormal input detection function of the three-phase encoder

NOTE

In this section, active level is indicated as high level.

25.2.2 Block Diagram

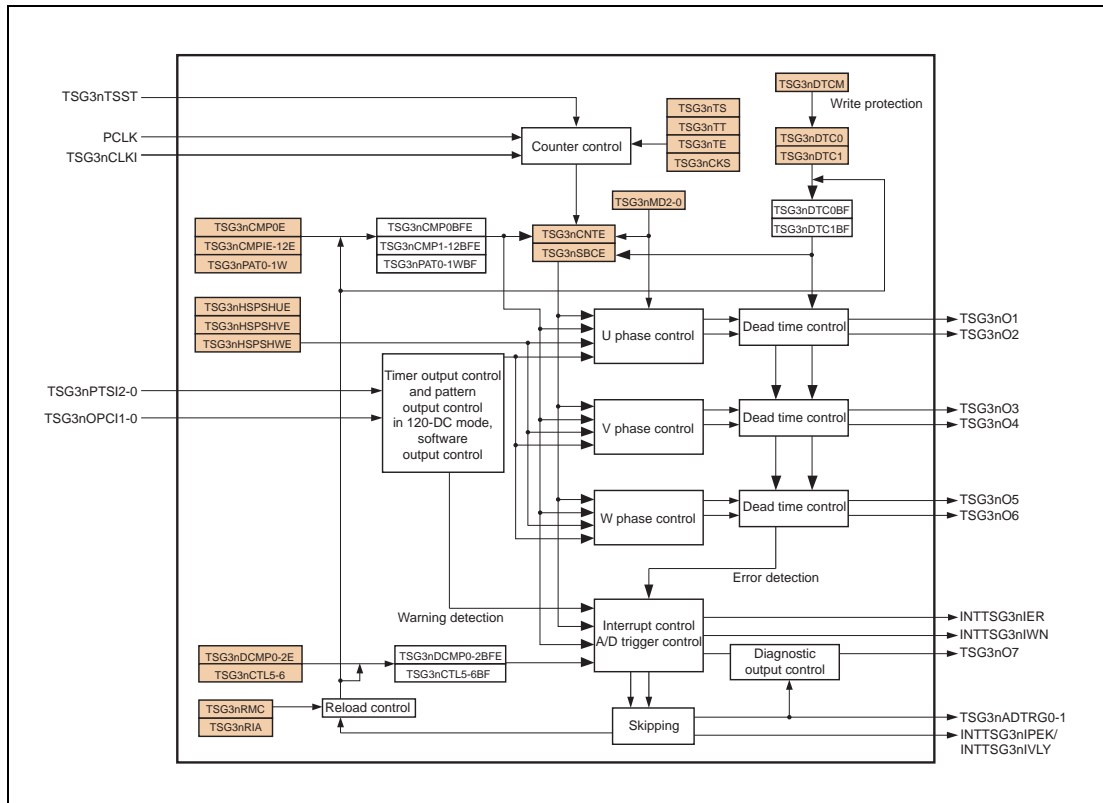


Figure 25.1 TSG3n Block Diagram

- TSG3nTSST: Simultaneous start trigger (input from PIC1A)

25.3 Registers

25.3.1 List of Registers

TSG3n registers are listed in the following table.

For <TSG3n_base>, see **Section 25.1.2, Register Base Address**

Table 25.7 List of Registers (1/3)

| Module Name | Register Name | Symbol | Address | Reload |
|-------------|---|-------------|---------------------------------|----------|
| TSG3n | TSG3n control register0 | TSG3nCTL0 | <TSG3n_base> + 208 _H | Disabled |
| TSG3n | TSG3n control register1 | TSG3nCTL1 | <TSG3n_base> + 20C _H | Disabled |
| TSG3n | TSG3n control register2 | TSG3nCTL2 | <TSG3n_base> + 78C _H | Enabled |
| TSG3n | TSG3n control register 3 | TSG3nCTL3 | <TSG3n_base> + 004 _H | Disabled |
| TSG3n | TSG3n control register 4 | TSG3nCTL4 | <TSG3n_base> + 07C _H | Enabled |
| TSG3n | TSG3n control register 5 | TSG3nCTL5 | <TSG3n_base> + 008 _H | Disabled |
| TSG3n | TSG3n control register 6 | TSG3nCTL6 | <TSG3n_base> + 00C _H | Disabled |
| TSG3n | TSG3n control register 7 | TSG3nCTL7 | <TSG3n_base> + 218 _H | Disabled |
| TSG3n | TSG3n control register 8 | TSG3nCTL8 | <TSG3n_base> + 21C _H | Disabled |
| TSG3n | TSG3n I/O control register 0 | TSG3nIOC0 | <TSG3n_base> + 200 _H | Disabled |
| TSG3n | TSG3n I/O control register 1 | TSG3nIOC1 | <TSG3n_base> + 204 _H | Disabled |
| TSG3n | TSG3n I/O control register 2 | TSG3nIOC2 | <TSG3n_base> + 000 _H | Disabled |
| TSG3n | TSG3n I/O control register 3 | TSG3nIOC3 | <TSG3n_base> + 074 _H | Enabled |
| TSG3n | TSG3n status register 0 | TSG3nSTR0 | <TSG3n_base> + 010 _H | Disabled |
| TSG3n | TSG3n status register 1 | TSG3nSTR1 | <TSG3n_base> + 014 _H | Disabled |
| TSG3n | TSG3n status register 2 | TSG3nSTR2 | <TSG3n_base> + 018 _H | Disabled |
| TSG3n | TSG3n status clear trigger register | TSG3nSTC | <TSG3n_base> + 01C _H | Disabled |
| TSG3n | TSG3n option register 0 | TSG3nOPT0 | <TSG3n_base> + 020 _H | Disabled |
| TSG3n | TSG3n option register 1 | TSG3nOPT1 | <TSG3n_base> + 024 _H | Disabled |
| TSG3n | TSG3n trigger register 0 | TSG3nTRG0 | <TSG3n_base> + 030 _H | Disabled |
| TSG3n | TSG3n trigger register 1 | TSG3nTRG1 | <TSG3n_base> + 034 _H | Disabled |
| TSG3n | TSG3n trigger register 2 | TSG3nTRG2 | <TSG3n_base> + 038 _H | Disabled |
| TSG3n | TSG3n counter read buffer register | TSG3nCNT | <TSG3n_base> + 028 _H | Disabled |
| TSG3n | TSG3n bit extended counter read buffer register | TSG3nCNTB | <TSG3n_base> + 1A0 _H | Disabled |
| TSG3n | TSG3n sub-counter read buffer register | TSG3nSBC | <TSG3n_base> + 02C _H | Disabled |
| TSG3n | TSG3n bit extended sub-counter read buffer register | TSG3nSBCB | <TSG3n_base> + 1A4 _H | Disabled |
| TSG3n | TSG3n compare register 0 | TSG3nCMP0 | <TSG3n_base> + 058 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 0 | TSG3nCMP0B | <TSG3n_base> + 14C _H | Enabled |
| TSG3n | TSG3n compare register 1, 2 | TSG3nCMP1W | <TSG3n_base> + 040 _H | Enabled |
| TSG3n | TSG3n compare register 5, 6 | TSG3nCMP5W | <TSG3n_base> + 044 _H | Enabled |
| TSG3n | TSG3n compare register 9, 10 | TSG3nCMP9W | <TSG3n_base> + 048 _H | Enabled |
| TSG3n | TSG3n compare register 3, 4 | TSG3nCMP3W | <TSG3n_base> + 04C _H | Enabled |
| TSG3n | TSG3n compare register 7, 8 | TSG3nCMP7W | <TSG3n_base> + 050 _H | Enabled |
| TSG3n | TSG3n compare register 11, 12 | TSG3nCMP11W | <TSG3n_base> + 054 _H | Enabled |
| TSG3n | TSG3n compare register 1 | TSG3nCMP1 | <TSG3n_base> + 080 _H | Enabled |
| TSG3n | TSG3n compare register 2 | TSG3nCMP2 | <TSG3n_base> + 084 _H | Enabled |
| TSG3n | TSG3n compare register 3 | TSG3nCMP3 | <TSG3n_base> + 098 _H | Enabled |
| TSG3n | TSG3n compare register 4 | TSG3nCMP4 | <TSG3n_base> + 09C _H | Enabled |

Table 25.7 List of Registers (2/3)

| Module Name | Register Name | Symbol | Address | Reload |
|-------------|---|--------------|---------------------------------|---------|
| TSG3n | TSG3n compare register 5 | TSG3nCMP5 | <TSG3n_base> + 088 _H | Enabled |
| TSG3n | TSG3n compare register 6 | TSG3nCMP6 | <TSG3n_base> + 08C _H | Enabled |
| TSG3n | TSG3n compare register 7 | TSG3nCMP7 | <TSG3n_base> + 0A0 _H | Enabled |
| TSG3n | TSG3n compare register 8 | TSG3nCMP8 | <TSG3n_base> + 0A4 _H | Enabled |
| TSG3n | TSG3n compare register 9 | TSG3nCMP9 | <TSG3n_base> + 090 _H | Enabled |
| TSG3n | TSG3n compare register 10 | TSG3nCMP10 | <TSG3n_base> + 094 _H | Enabled |
| TSG3n | TSG3n compare register 11 | TSG3nCMP11 | <TSG3n_base> + 0A8 _H | Enabled |
| TSG3n | TSG3n compare register 12 | TSG3nCMP12 | <TSG3n_base> + 0AC _H | Enabled |
| TSG3n | TSG3n bit extended compare register 1 | TSG3nCMP1E | <TSG3n_base> + 17C _H | Enabled |
| TSG3n | TSG3n bit extended compare register 2 | TSG3nCMP2E | <TSG3n_base> + 178 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 3 | TSG3nCMP3E | <TSG3n_base> + 164 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 4 | TSG3nCMP4E | <TSG3n_base> + 160 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 5 | TSG3nCMP5E | <TSG3n_base> + 174 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 6 | TSG3nCMP6E | <TSG3n_base> + 170 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 7 | TSG3nCMP7E | <TSG3n_base> + 15C _H | Enabled |
| TSG3n | TSG3n bit extended compare register 8 | TSG3nCMP8E | <TSG3n_base> + 158 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 9 | TSG3nCMP9E | <TSG3n_base> + 16C _H | Enabled |
| TSG3n | TSG3n bit extended compare register 10 | TSG3nCMP10E | <TSG3n_base> + 168 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 11 | TSG3nCMP11E | <TSG3n_base> + 154 _H | Enabled |
| TSG3n | TSG3n bit extended compare register 12 | TSG3nCMP12E | <TSG3n_base> + 150 _H | Enabled |
| TSG3n | TSG3n diagnostic output compare register 0, 1 | TSG3nDCMP0W | <TSG3n_base> + 05C _H | Enabled |
| TSG3n | TSG3n diagnostic output compare register 2 | TSG3nDCMP2 | <TSG3n_base> + 060 _H | Enabled |
| TSG3n | TSG3n bit extended diagnostic output compare register 0 | TSG3nDCMP0E | <TSG3n_base> + 148 _H | Enabled |
| TSG3n | TSG3n bit extended diagnostic output compare register 1 | TSG3nDCMP1E | <TSG3n_base> + 144 _H | Enabled |
| TSG3n | TSG3n bit extended diagnostic output compare register 2 | TSG3nDCMP2E | <TSG3n_base> + 140 _H | Enabled |
| TSG3n | TSG3n pattern register 0 | TSG3nPAT0W | <TSG3n_base> + 064 _H | Enabled |
| TSG3n | TSG3n pattern register 1 | TSG3nPAT1W | <TSG3n_base> + 068 _H | Enabled |
| TSG3n | TSG3n dead time control register 0 | TSG3nDTC0W | <TSG3n_base> + 06C _H | Enabled |
| TSG3n | TSG3n dead time control register 1 | TSG3nDTC1W | <TSG3n_base> + 070 _H | Enabled |
| TSG3n | TSG3n HT-PWM U phase compare register | TSG3nCMPU | <TSG3n_base> + 0B0 _H | Enabled |
| TSG3n | TSG3n HT-PWM V phase compare register | TSG3nCMPV | <TSG3n_base> + 0B4 _H | Enabled |
| TSG3n | TSG3n HT-PWM W phase compare register | TSG3nCMPW | <TSG3n_base> + 0B8 _H | Enabled |
| TSG3n | TSG3n bit extended HT-PWM U phase compare register | TSG3nCMPUE | <TSG3n_base> + 188 _H | Enabled |
| TSG3n | TSG3n bit extended HT-PWM V phase compare register | TSG3nCMPVE | <TSG3n_base> + 184 _H | Enabled |
| TSG3n | TSG3n bit extended HT-PWM W phase compare register | TSG3nCMPWE | <TSG3n_base> + 180 _H | Enabled |
| TSG3n | TSG3n SP-PWM U phase active width register | TSG3nUPW | <TSG3n_base> + 0BC _H | Enabled |
| TSG3n | TSG3n SP-PWM V phase active width register | TSG3nVPW | <TSG3n_base> + 0C0 _H | Enabled |
| TSG3n | TSG3n SP-PWM W phase active width register | TSG3nWPW | <TSG3n_base> + 0C4 _H | Enabled |
| TSG3n | TSG3n bit extended SP-PWM U phase active width register | TSG3nUPWE | <TSG3n_base> + 198 _H | Enabled |
| TSG3n | TSG3n bit extended SP-PWM V phase active width register | TSG3nVPWE | <TSG3n_base> + 194 _H | Enabled |
| TSG3n | TSG3n bit extended SP-PWM W phase active width register | TSG3nWPWE | <TSG3n_base> + 190 _H | Enabled |
| TSG3n | TSG3n HSP-PWM W phase shift register | TSG3nHSPSHWE | <TSG3n_base> + 120 _H | Enabled |

Table 25.7 List of Registers (3/3)

| Module Name | Register Name | Symbol | Address | Reload |
|-------------|--|--------------|---------------------------------|----------|
| TSG3n | TSG3n HSP-PWM V phase shift register | TSG3nHSPSHVE | <TSG3n_base> + 124 _H | Enabled |
| TSG3n | TSG3n HSP-PWM U phase shift register | TSG3nHSPSHUE | <TSG3n_base> + 128 _H | Enabled |
| TSG3n | TSG3n HSP-PWM W phase compare register | TSG3nHSPCMWE | <TSG3n_base> + 12C _H | Enabled |
| TSG3n | TSG3n HSP-PWM V phase compare register | TSG3nHSPCMVE | <TSG3n_base> + 130 _H | Enabled |
| TSG3n | TSG3n HSP-PWM U phase compare register | TSG3nHSPCMUE | <TSG3n_base> + 134 _H | Enabled |
| TSG3n | TSG3n dead time protection register | TSG3nDTPR | <TSG3n_base> + 210 _H | Disabled |

25.3.2 TSG3nCTL0 — TSG3n Control Register 0

This register specifies the pulse width for the diagnostic output and operating mode of the TSG3n.

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> + 208_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|----------|---|--------------|-----|-----|
| | — | — | — | TSG3nDWD | — | TSG3nMD[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R | R/W | R/W | R/W |

Table 25.8 TSG3nCTL0 Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|--------------|--|--|-----------|-----------|------------|---|---|---|----------|---|---|---|----------------------|---|---|---|-------------------------------|---|---|---|-------------|---|---|---|--|------------------|--|--|--------------------|
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | TSG3nDWD | <p>Selects the pulse width for the diagnostic output.</p> <p>0: The output pulse width is set to 8 clocks.</p> <p>1: The output pulse width is set to 16 clocks.</p> <p>The setting of this bit is valid when diagnostic output is enabled (TSG3nIOC1.TSG3nTGS = 1).</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 to 0 | TSG3nMD[2:0] | <p>Selects timer mode</p> <table border="1"> <thead> <tr> <th>TSG3n MD2</th> <th>TSG3n MD1</th> <th>TSG3n MD0</th> <th>Timer Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>PWM mode</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>HT-PWM mode (HT-PWM)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Shift pulse PWM mode (SP-PWM)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>120-DC mode</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>High-accuracy shift pulse PWM mode (HSP-PWM)</td> </tr> <tr> <td colspan="3">Other than above</td> <td>Setting prohibited</td> </tr> </tbody> </table> | TSG3n MD2 | TSG3n MD1 | TSG3n MD0 | Timer Mode | 0 | 0 | 0 | PWM mode | 0 | 0 | 1 | HT-PWM mode (HT-PWM) | 0 | 1 | 0 | Shift pulse PWM mode (SP-PWM) | 0 | 1 | 1 | 120-DC mode | 1 | 0 | 0 | High-accuracy shift pulse PWM mode (HSP-PWM) | Other than above | | | Setting prohibited |
| TSG3n MD2 | TSG3n MD1 | TSG3n MD0 | Timer Mode | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | PWM mode | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | HT-PWM mode (HT-PWM) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | Shift pulse PWM mode (SP-PWM) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 120-DC mode | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | High-accuracy shift pulse PWM mode (HSP-PWM) | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Other than above | | | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | | | | | |

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer

25.3.3 TSG3nCTL1 — TSG3n Control Register 1

This register controls the flags of TSG3n.

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 20C_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|----------------|
| | — | — | — | — | — | — | TSG3nTBA2 | TSG3nTBA1 | TSG3nTBA0 | TSG3nPPC | TSG3nPEC | TSG3nTDC | TSG3nNDC | TSG3nPRC | TSG3nPTC | TSG3nPTC [1:0] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 25.9 TSG3nCTL1 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 15 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | TSG3nTBA2 | <p>Enables or disables detection of the simultaneous active states of the TSG3nO5 and TSG3nO6 pins.</p> <p>0: Disables detection of simultaneous active states of the TSG3nO5 and TSG3nO6 pins.</p> <p>1: Enables detection of simultaneous active states of the TSG3nO5 and TSG3nO6 pins.</p> <p>If the simultaneous active state is detected when the TSG3nIOC1.TSG3nEOC and TSG3nTBA2 bits are 1, the positive phase and inverse phase simultaneous active state detection flag 2 (TSG3nTBF2) is set to 1, and an error interrupt (INTTSG3nIER) is generated.</p> |
| 8 | TSG3nTBA1 | <p>Enables or disables detection of the simultaneous active states of the TSG3nO3 and TSG3nO4 pins.</p> <p>0: Disables detection of simultaneous active states of the TSG3nO3 and TSG3nO4 pins.</p> <p>1: Enables detection of simultaneous active states of the TSG3nO3 and TSG3nO4 pins.</p> <p>If the simultaneous active state is detected when the TSG3nIOC1.TSG3nEOC and TSG3nTBA1 bits are 1, the positive phase and inverse phase simultaneous active state detection flag 1 (TSG3nTBF1) is set to 1, and an error interrupt (INTTSG3nIER) is generated.</p> |
| 7 | TSG3nTBA0 | <p>Enables or disables detection of the simultaneous active states of the TSG3nO1 and TSG3nO2 pins.</p> <p>0: Disables detection of simultaneous active states of the TSG3nO1 and TSG3nO2 pins.</p> <p>1: Enables detection of simultaneous active states of the TSG3nO1 and TSG3nO2 pins.</p> <p>If the simultaneous active state is detected when the TSG3nIOC1.TSG3nEOC and TSG3nTBA0 bits are 1, the positive phase and inverse phase simultaneous active state detection flag 0 (TSG3nTBF0) is set to 1, and an error interrupt (INTTSG3nIER) is generated.</p> |
| 6 | TSG3nPPC | <p>Enables or disables detection of the pattern phase difference (TSG3nSTR2.TSG3nPPF) between the TSG3nPTSI2-0 and TSG3nOPF2-0.</p> <p>0: Disables detection of I/O pattern difference.</p> <p>1: Enables detection of I/O pattern difference</p> |
| 5 | TSG3nPEC | <p>Enables or disables detection of the pattern error (TSG3nSTR2.TSG3nPEF) of the TSG3nPTSI2-0 pins.</p> <p>0: Disables detection of the pattern error of the TSG3nPTSI2-0 pins.</p> <p>1: Enables detection of the pattern error of the TSG3nPTSI2-0 pins.</p> |

Table 25.9 TSG3nCTL1 Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|---------------|--|------------|------------|----------|---|---|--|---|---|--|---|---|---|---|---|--|
| 4 | TSG3nTDC | Enables or disables detection of the simultaneous trigger (TSG3nSTR2.TSG3nTDF) of the TSG3nOPCI0 and TSG3nOPCI1. 0: Disables detection of the simultaneous trigger of the TSG3nOPCI0 and TSG3nOPCI1. 1: Enables detection of the simultaneous trigger of the TSG3nOPCI0 and TSG3nOPCI1. | | | | | | | | | | | | | | | |
| 3 | TSG3nNDC | Enables or disables detection of the noise generation (two or more pins change simultaneously) (TSG3nSTR2.TSG3nNDF) on the TSG3nPTSI2-0 pins. 0: Disables detection of the noise generation on the TSG3nPTSI2-0 pins. 1: Enables detection of the noise generation on the TSG3nPTSI2-0 pins. | | | | | | | | | | | | | | | |
| 2 | TSG3nPRC | Enables or disables detection of the reversal of the pattern (TSG3nSTR2.TSG3nPRF) of the TSG3nPTSI2-0 pins. 0: Disables detection of the reversal of the pattern of the TSG3nPTSI2-0 pins. 1: Enables detection of the reversal of the pattern of the TSG3nPTSI2-0 pins. | | | | | | | | | | | | | | | |
| 1, 0 | TSG3nPTC[1:0] | Enables or disables detection of an abnormal toggle (TSG3nSTR2.TSG3nPTF) of the TSG3nPTSI2-0 pins between TSG3nOPCI1 and TSG3nOPCI0 triggers <table border="1" data-bbox="678 788 1417 1079"> <thead> <tr> <th>TSG3n PTC1</th> <th>TSG3n PTC0</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Disables detection of an abnormal toggle of the TSG3nPTSI2-0 pins.</td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. When an abnormal toggle is detected, the pattern output switch trigger is automatically switched from trigger switch to pattern switch (TSG3nPOT is switched from 1 to 0.)</td> </tr> </tbody> </table> | TSG3n PTC1 | TSG3n PTC0 | Function | 0 | 0 | Disables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. | 0 | 1 | | 1 | 0 | Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. | 1 | 1 | Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. When an abnormal toggle is detected, the pattern output switch trigger is automatically switched from trigger switch to pattern switch (TSG3nPOT is switched from 1 to 0.) |
| TSG3n PTC1 | TSG3n PTC0 | Function | | | | | | | | | | | | | | | |
| 0 | 0 | Disables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 0 | Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. | | | | | | | | | | | | | | | |
| 1 | 1 | Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. When an abnormal toggle is detected, the pattern output switch trigger is automatically switched from trigger switch to pattern switch (TSG3nPOT is switched from 1 to 0.) | | | | | | | | | | | | | | | |

CAUTIONS

1. If TSG3nDTC0 or TSG3nDTC1 is set to 0000_H (without dead time), the TSG3nTBA2-0 bits should be set to 0.
2. This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer

25.3.4 TSG3nCTL2 — TSG3n Control Register 2

This register selects a count clock for TSG3n.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 078_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3n CKS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 25.10 TSG3nCTL2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TSG3nCKS | Selects a count clock 0: Selects PCLK as a count clock. 1: Counted by PCLK when clock enable input (TSG3nCLKI) is high level. |

CAUTION

Set the TSG3nCTL2.TSG3nCKS bit to 0 in HT-PWM mode and HSP-PWM mode.

25.3.5 TSG3nCTL3 — TSG3n Control Register 3

This register selects the rewrite method of the compare registers.

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> + 004_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|----------|----------|
| | — | — | — | — | — | — | TSG3nRIA | TSG3nRMC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 25.11 TSG3nCTL3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | TSG3nRIA | <p>Selects the reload timing of the compare register values.</p> <p>0: The reload timing is set to peak reload timing (set by TSG3nCTL4.TSG3nPRE) and valley reload timing (set by TSG3nCTL4.TSG3nVRE).</p> <p>1: The reload timing is set to peak interrupt timing and valley interrupt timing.</p> <p>The setting of this bit is valid in reload mode (TSG3nRMC = 0).</p> |
| 0 | TSG3nRMC | <p>Selects the transfer timing of the compare register values.</p> <p>0: Reload mode (simultaneous rewrite) Writing to registers to be reloaded enables reloading and the register values are rewritten simultaneously at the next reload timing. Writing to any register other than registers to be reloaded does not enable reloading. For the register to be reloaded, see Section 25.3.1, List of Registers.</p> <p>1: Anytime rewrite mode The compare registers are rewritten independently. Whenever a value is written to the compare register, the written value is reflected immediately. TSG3nRSF is cleared. Do not set TSG3nRMC to 1 when operated in 120-DC mode or in HSP-PWM mode.</p> |

25.3.6 TSG3nCTL4 — TSG3n Control Register 4

This register enables or disables generation of a peak interrupt and a valley interrupt, and the reload timing.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 07C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|--------------|--------------|--------------|--------------|-----------------|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TSG3n PRE | TSG3n VRE | TSG3n PIE | TSG3n VIE | TSG3nRCC[04:00] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 25.12 TSG3nCTL4 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | TSG3nPRE | <p>Enables or disables the peak reload timing.</p> <p>0: Disables reload operation at the peak timing of the 18-bit counter.</p> <p>1: Enables reload operation at the peak timing of the 18-bit counter.</p> <ul style="list-style-type: none"> The peak reload timing means the peak timing of the 18-bit counter in HT-PWM mode and the clear timing of the 18-bit counter by compare match in any mode other than HT-PWM mode. When the reload operation at the peak timing of the 18-bit counter is disabled (TSG3nPRE = 0), reload is not executed in any mode other than HT-PWM mode. |
| 7 | TSG3nVRE | <p>Enables or disables the valley reload timing.</p> <p>0: Disables reload operation at the valley timing of the 18-bit counter.</p> <p>1: Enables reload operation at the valley timing of the 18-bit counter.</p> <p>The setting of this bit is valid only in HT-PWM mode.</p> |
| 6 | TSG3nPIE | <p>Enables or disables generation of a peak interrupt (INTTSG3nIPEK).</p> <p>0: Disables generation of a peak interrupt (INTTSG3nIPEK) at the peak timing of the 18-bit counter. Interrupts are not skipped.</p> <p>1: Enables generation of a peak interrupt (INTTSG3nIPEK) at the peak timing of the 18-bit counter. Interrupts are skipped.</p> |
| 5 | TSG3nVIE | <p>Enables or disables generation of a valley interrupt (INTTSG3nIVLY).</p> <p>0: Disables generation of a valley interrupt (INTTSG3nIVLY) at the valley timing of the 18-bit counter. Interrupts are not skipped.</p> <p>1: Enables generation of a valley interrupt (INTTSG3nIVLY) at the valley timing of the 18-bit counter. Interrupts are skipped.</p> <p>The setting of this bit is valid only in HT-PWM mode.</p> |

Table 25.12 TSG3nCTL4 Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------|---------------------|---|----------------|----------------|----------------------|----------------|----------------|------------------|---|---|---|---|---|----------------------|---|---|---|---|---|-----|---|---|---|---|---|-----|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|------|---|---|---|---|---|------|---|---|---|---|---|------|
| 4 to 0 | TSG3nRCC [04:00] | Specifies the skipping rate of the interrupts (INTTSG3nIPEK and INTTSG3nIVLY) and reload. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>TSG3n RCC04</th> <th>TSG3n RCC03</th> <th>TSG3n RCC02</th> <th>TSG3n RCC01</th> <th>TSG3n RCC00</th> <th>Skipping Rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Skipping Disabled</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1/2</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1/3</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1/4</td> </tr> <tr> <td>:</td> <td>:</td> <td>:</td> <td>:</td> <td>:</td> <td>:</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1/30</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>1/31</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1/32</td> </tr> </tbody> </table> | TSG3n RCC04 | TSG3n RCC03 | TSG3n RCC02 | TSG3n RCC01 | TSG3n RCC00 | Skipping Rate | 0 | 0 | 0 | 0 | 0 | Skipping Disabled | 0 | 0 | 0 | 0 | 1 | 1/2 | 0 | 0 | 0 | 1 | 0 | 1/3 | 0 | 0 | 0 | 1 | 1 | 1/4 | : | : | : | : | : | : | 1 | 1 | 1 | 0 | 1 | 1/30 | 1 | 1 | 1 | 1 | 0 | 1/31 | 1 | 1 | 1 | 1 | 1 | 1/32 |
| TSG3n RCC04 | TSG3n RCC03 | TSG3n RCC02 | TSG3n RCC01 | TSG3n RCC00 | Skipping Rate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | Skipping Disabled | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 1/2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 1/3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1 | 1/4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| : | : | : | : | : | : | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 0 | 1 | 1/30 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 0 | 1/31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1/32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

When a write access is made (including a write of the same value to TSG3nRCC04-TSG3nRCC00) to TSG3nCTL4 during timer operation (TSG3nSTR0.TSG3nTE = 1), the interrupt skipping counter is cleared

25.3.7 TSG3nCTL5 — TSG3n Control Register5

This register controls A/D conversion trigger output (TSG3nADTRG0).

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> +008_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|-----|
| | — | — | — | — | TSG3nACC [01:00] | TSG3n AT09 | TSG3n AT08 | TSG3n AT07 | TSG3n AT06 | TSG3n AT05 | TSG3n AT04 | TSG3n AT03 | TSG3n AT02 | TSG3n AT01 | TSG3n AT00 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 25.13 TSG3nCTL5 Register Contents (1/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|---------------------|--|------------|------------|---------------|---|---|-------------------|---|---|-----|---|---|-----|---|---|-----|
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 11, 10 | TSG3nACC [01:00] | Specifies the skipping rate of the A/D conversion trigger (TSG3nADTRG0). <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TSG3nACC01</th> <th>TSG3nACC00</th> <th>Skipping Rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Skipping Disabled</td> </tr> <tr> <td>0</td> <td>1</td> <td>1/2</td> </tr> <tr> <td>1</td> <td>0</td> <td>1/4</td> </tr> <tr> <td>1</td> <td>1</td> <td>1/8</td> </tr> </tbody> </table> <p>When a write access is made (including a write of the same value to TSG3nACC01 and TSG3nACC00) to TSG3nCTL5 during timer operation (TSG3nSTR0.TSG3nTE = 1), the interrupt skipping counter is cleared.</p> | TSG3nACC01 | TSG3nACC00 | Skipping Rate | 0 | 0 | Skipping Disabled | 0 | 1 | 1/2 | 1 | 0 | 1/4 | 1 | 1 | 1/8 |
| TSG3nACC01 | TSG3nACC00 | Skipping Rate | | | | | | | | | | | | | | | |
| 0 | 0 | Skipping Disabled | | | | | | | | | | | | | | | |
| 0 | 1 | 1/2 | | | | | | | | | | | | | | | |
| 1 | 0 | 1/4 | | | | | | | | | | | | | | | |
| 1 | 1 | 1/8 | | | | | | | | | | | | | | | |
| 9 | TSG3nAT09 | Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the (peak) timing when the 18-bit sub-counter switches from incrementing to decrementing. <ul style="list-style-type: none"> 0: Disables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. <ul style="list-style-type: none"> • The TSG3nAT09 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT09 bit should be set to 0. • Do not set the TSG3nAT09 bit to 1 when TSG3nDTC0W is not 0000_H and TSG3nDTC1W is 0000_H. A/D conversion trigger is not generated at the peak timing of the 18-bit sub-counter even if set so. | | | | | | | | | | | | | | | |
| 8 | TSG3nAT08 | Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the (valley) timing when the 18-bit sub-counter switches from decrementing to incrementing. <ul style="list-style-type: none"> 0: Disables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. <ul style="list-style-type: none"> • The TSG3nAT08 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT08 bit should be set to 0. • Do not set the TSG3nAT08 bit to 1 when TSG3nDTC0W is 0000_H and TSG3nDTC1W is not 0000_H. A/D conversion trigger is not generated at the valley timing of the 18-bit sub-counter even if set so. | | | | | | | | | | | | | | | |

Table 25.13 TSG3nCTL5 Register Contents (2/3)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 | TSG3nAT07 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.</p> |
| 6 | TSG3nAT06 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value.</p> |
| 5 | TSG3nAT05 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.</p> |
| 4 | TSG3nAT04 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value.</p> |
| 3 | TSG3nAT03 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.</p> |

Table 25.13 TSG3nCTL5 Register Contents (3/3)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 2 | TSG3nAT02 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value.</p> <p>1: Enables generation of the A/D conversion trigger at the timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E.</p> |
| 1 | TSG3nAT01 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the timing (peak interrupt) when the 18-bit counter switches from incrementing to decrementing.</p> <p>0: Disables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped.</p> <p>1: Enables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped.</p> |
| 0 | TSG3nAT00 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the timing (valley interrupt) when the 18-bit counter switches from decrementing to incrementing.</p> <p>0: Disables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped.</p> <p>1: Enables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.</p> |

25.3.8 TSG3nCTL6 — TSG3n Control Register 6

This register controls the A/D conversion trigger output (TSG3nADTRG1).

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 00C_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|-----|
| | — | — | — | — | TSG3nACC [11:10] | TSG3n AT19 | TSG3n AT18 | TSG3n AT17 | TSG3n AT16 | TSG3n AT15 | TSG3n AT14 | TSG3n AT13 | TSG3n AT12 | TSG3n AT11 | TSG3n AT10 | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 25.14 TSG3nCTL6 Register Contents (1/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|---------------------|--|------------|------------|---------------|---|---|-------------------|---|---|-----|---|---|-----|---|---|-----|
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 11, 10 | TSG3nACC [11:10] | Specifies the skipping rate of the A/D conversion trigger (TSG3nADTRG1). <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TSG3nACC11</th> <th>TSG3nACC10</th> <th>Skipping Rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Skipping Disabled</td> </tr> <tr> <td>0</td> <td>1</td> <td>1/2</td> </tr> <tr> <td>1</td> <td>0</td> <td>1/4</td> </tr> <tr> <td>1</td> <td>1</td> <td>1/8</td> </tr> </tbody> </table> <p>When a write access is made (including a write of the same value to TSG3nACC11 and TSG3nACC10) to TSG3nCTL6 during timer operation (TSG3nSTR0.TSG3nTE = 1), the skipping counter is cleared.</p> | TSG3nACC11 | TSG3nACC10 | Skipping Rate | 0 | 0 | Skipping Disabled | 0 | 1 | 1/2 | 1 | 0 | 1/4 | 1 | 1 | 1/8 |
| TSG3nACC11 | TSG3nACC10 | Skipping Rate | | | | | | | | | | | | | | | |
| 0 | 0 | Skipping Disabled | | | | | | | | | | | | | | | |
| 0 | 1 | 1/2 | | | | | | | | | | | | | | | |
| 1 | 0 | 1/4 | | | | | | | | | | | | | | | |
| 1 | 1 | 1/8 | | | | | | | | | | | | | | | |
| 9 | TSG3nAT19 | Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the (peak) timing when the 18-bit sub-counter switches from incrementing to decrementing. <ul style="list-style-type: none"> 0: Disables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. <ul style="list-style-type: none"> • The TSG3nAT19 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT19 bit should be set to 0. • Do not set the TSG3nAT19 bit to 1 when TSG3nDTC0W is not 0000_H and TSG3nDTC1W is 0000_H. A/D conversion trigger is not generated at the peak timing of the 18-bit sub-counter even if set so. | | | | | | | | | | | | | | | |
| 8 | TSG3nAT18 | Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the (valley) timing when the 18-bit sub-counter switches from decrementing to incrementing. <ul style="list-style-type: none"> 0: Disables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. <ul style="list-style-type: none"> • The TSG3nAT18 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT18 bit should be set to 0. • Do not set the TSG3nAT18 bit to 1 when TSG3nDTC0W is 0000_H and TSG3nDTC1W is not 0000_H. A/D conversion trigger is not generated at the valley timing of the 18-bit sub-counter even if set so. | | | | | | | | | | | | | | | |

Table 25.14 TSG3nCTL6 Register Contents (2/3)

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 | TSG3nAT17 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during decrementation and the TSG3nDCMP2E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value.</p> <p>1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.</p> |
| 6 | TSG3nAT16 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value.</p> <p>1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value.</p> |
| 5 | TSG3nAT15 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value.</p> <p>1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.</p> |
| 4 | TSG3nAT14 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value.</p> <p>1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value.</p> |
| 3 | TSG3nAT13 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value.</p> <p>1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.</p> |

Table 25.14 TSG3nCTL6 Register Contents (3/3)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 2 | TSG3nAT12 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E.</p> <p>0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value.</p> <p>1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value.</p> |
| 1 | TSG3nAT11 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the timing (peak interrupt) when the 18-bit counter switches from incrementing to decrementing.</p> <p>0: Disables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped.</p> <p>1: Enables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped.</p> |
| 0 | TSG3nAT10 | <p>Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the timing (valley interrupt) when the 18-bit counter switches from incrementing to decrementing.</p> <p>0: Disables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped.</p> <p>1: Enables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped.</p> <p>This bit can be set to 1 only in HT-PWM mode. In other modes, this bit must be set to 0.</p> |

25.3.9 TSG3nCTL7 — TSG3n Control Register 7

This register sets the level of PWM output from the TSG3O1 to TSG3O6 pins at operation start (TSG3nTE is changed from 0 to 1) and at operation restart in SP-PWM mode.

This register can be written only when SP-PWM mode is selected (TSG3nMD2 to TSG3nMD0 = 010) and the timer is stopped (TSG3nTE = 0). Do not rewrite this register in other modes (PWM mode, HT-PWM mode, 120-DC mode, HSP-PWM mode) or while the timer is operating (TSG3nTE = 1).

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> + 218_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|-----------------|-----------------|-----------------|
| | — | — | — | — | — | TSG3n SPSTL2 | TSG3n SPSTL1 | TSG3n SPSTL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 25.15 TSG3nCTL7 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 7 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | TSG3nSPSTL2 | SP-PWM Mode Start Level Control Bit 2 0: TSG3nO5 (W phase) is cleared and TSG3nO6 (WB phase) is set at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. 1: TSG3nO5 (W phase) is set and TSG3nO6 (WB phase) is cleared at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. |
| 1 | TSG3nSPSTL1 | SP-PWM Mode Start Level Control Bit 1 0: TSG3nO3 (V phase) is cleared and TSG3nO4 (VB phase) is set at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. 1: TSG3nO3 (V phase) is set and TSG3nO4 (VB phase) is cleared at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. |
| 0 | TSG3nSPSTL0 | SP-PWM Mode Start Level Control Bit 0 0: TSG3nO1 (U phase) is cleared and TSG3nO2 (UB phase) is set at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. 1: TSG3nO3 (U phase) is set and TSG3nO1 (UB phase) is cleared at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. |

NOTE

The settings of bits TSG3nSPSTL2 to TSG3nSPSTL0 affect output on the TSG3nO1 to TSG3nO6 pins when operation starts or is restarted. The set dead time is always inserted at these times.

25.3.10 TSG3nCTL8 — TSG3n Control Register 8

This register specifies timer output timing when input patterns are changed in 120-DC mode.

This register can be written only when 120-DC mode is selected (TSG3nMD2-0 = 011) and the timer is stopped (TSG3nTE = 0).

Do not rewrite this register in other modes (PWM mode, SP-PMW mode, HT-PWM mode, HSP-PWM mode) or while the timer is operating (TSG3nTE = 1).

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> +21C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|------------------|
| | — | — | — | — | — | — | — | TSG3n S120DCO |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 25.16 TSG3nCTL8 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TSG3n S120DCO | 120-DC Mode Control Bit 0 0: When the input patterns are changed while 120-DC mode is selected, the main counter (TSG3nCnTE) is cleared and the change of input patterns is immediately reflected to timer output. 1: When the input patterns are changed while 120-DC mode is selected, the change of input patterns is reflected to timer output after a match of the main counter (TSG3nCnTE) with TSG3nCMP0E (from the next timer period). |

CAUTION

When TSG3nS120DCO is set to 1 in 120DC mode, set the TSG3nOPT0.TSG3nSOC.

The settings of the TSG3nOPT0.TSG3nSTE and TSG3nOPT0.TSG3nPOT bits must not be changed while the timer is operating (TSG3nSTR0.TSG3nTE = 1).

25.3.11 TSG3nIOC0 — TSG3n I/O Control Register0

This register controls the timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> +200_H

Value after reset: 7E_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|-----------|-----------|-----------|-----------|-----------|-----------|---|
| | — | TSG3nTOE6 | TSG3nTOE5 | TSG3nTOE4 | TSG3nTOE3 | TSG3nTOE2 | TSG3nTOE1 | — |
| Value after reset | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 25.17 TSG3nIOC0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 1 | TSG3nTOE6 to TSG3nTOE1 | Enables or disables controlling of TSG3nO6 to TSG3nO1 by TSG3nIOC2. When these bits are 1, rewriting of TSG3nIOC2 is ignored. 0: Controlling is enabled. 1: Controlling is disabled. |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

25.3.12 TSG3nIOC1 — TSG3n I/O Control Register1

This register controls the timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> + 204_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|----------|----------|----------|----------|----------|
| | — | — | — | TSG3nPTS | TSG3nEOC | TSG3nWOC | TSG3nTGS | TSG3nTOS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 25.18 TSG3nIOC1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | TSG3nPTS | Enables or disables output of the edge detection signal (TSG3nPTE) of TSG3nPTSI0 to TSG3nPTSI2 and two-phase encoder count signal (TSG3nPEC). 0: Disables output of the toggle signal by edge detection of TSG3nPTSI0 to TSG3nPTSI2. 1: Enables output of the toggle signal by edge detection of TSG3nPTSI0 to TSG3nPTSI2. |
| 3 | TSG3nEOC | Enables or disables detection of the error condition at the motor control. 0: Disables generation of an error interrupt (INTTSG3nIER). 1: Enables generation of an error interrupt (INTTSG3nIER). For details on controlling the error interrupt, see Section 25.4.6.1, Error Interrupt Function . |
| 2 | TSG3nWOC | Enables or disables detection of the warning condition at the motor control. 0: Disables generation of a warning interrupt (INTTSG3nIWN). 1: Enables generation of a warning interrupt (INTTSG3nIWN). For details on the controlling generation of warning interrupt, see Section 25.4.6.2, Warning Interrupt Function . |
| 1 | TSG3nTGS | Selects the A/D conversion trigger diagnostic output (TSG3nO7) signal. 0: Selects A/D conversion trigger output. 1: Selects diagnostic output. |
| 0 | TSG3nTOS | Selects the timer counter increment/decrement status output (TSG3nO0) signal. 0: Outputs the up/down count flag of the 18-bit counter. 1: Outputs the up/down count flag of the 18-bit sub-counter. <ul style="list-style-type: none"> • When TSG3nTOS is 0, the status of TSG3nSTR0.TSG3nCUF is output to TSG3nO0. When TSG3nTOS is 1, the status of TSG3nSTR0.TSG3nSUF is output to TSG3nO0. • The setting of this bit is valid only in HT-PWM mode. |

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

25.3.13 TSG3nIOC2 — TSG3n I/O Control Register2

This register controls the timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 000_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----------|----------|----------|----------|----------|----------|---|---|----------|----------|----------|----------|----------|----------|---|
| | — | TSG3nOL6 | TSG3nOL5 | TSG3nOL4 | TSG3nOL3 | TSG3nOL2 | TSG3nOL1 | — | — | TSG3nTO6 | TSG3nTO5 | TSG3nTO4 | TSG3nTO3 | TSG3nTO2 | TSG3nTO1 | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 25.19 TSG3nIOC2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 14 to 9 | TSG3nOL6 to TSG3nOL1 | Specifies the active level of TSG3nO6 to TSG3nO1 outputs. 0: Active level is high level 1: Active level is low level |
| 8 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 1 | TSG3nTO6 to TSG3nTO1 | Specifies the latch level of the output buffer of the TSG3nO6 to TSG3nO1. 0: Latch level of output buffer is low level 1: Latch level of output buffer is high level |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

CAUTION

When the counting is stopped (TSG3nSTR0.TSG3nTE = 0), the TSG3nO1-6 pins maintain their previous output states. The output level should be changed by setting the TSG3nIOC0.TSG3nTOEm bit to 0, using the TSG3nTOm bit. This register can be rewritten when TSG3nIOC0.TSG3nTOEm = 0 (m = 1 to 6).

NOTE

While the timer is stopped (TSG3nSTR0.TSG3nTE = 0) and control of TSG3nOm by rewriting TSG3nIOC2 is enabled (TSG3nIOC0.TSG3nTOEm = 0), TSG3nOLm and TSG3nTOm of TSG3nIOC2 select the output level for the corresponding TSG3nOm output as listed in the table below.

| TSG3nOLm | TSG3nTOm | Output level of TSG3nOm |
|----------|----------|-------------------------|
| 0 | 0 | Low level |
| 0 | 1 | High level |
| 1 | 0 | High level |
| 1 | 1 | Low level |

25.3.14 TSG3nIOC3 — TSG3n I/O Control Register3

This register controls timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 074_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---------------|---------------|---------------|---------------|---------------|---------------|---|
| | — | — | — | — | — | — | — | — | — | TSG3n TOL6 | TSG3n TOL5 | TSG3n TOL4 | TSG3n TOL3 | TSG3n TOL2 | TSG3n TOL1 | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 25.20 TSG3nIOC3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------------|--|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 1 | TSG3nTOL6 to TSG3nTOL1 | Controls the set/clear level of output. 0: Outputs the normal level. 1: Outputs the reversed level. Setting of this bit is reflected at the start of output. The change of the output level is reflected at the next compare match timing after the change. |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

CAUTION

TSG3nTOL6 to TSG3nTOL1 should be set to 0 in HT-PWM mode and HSP-PWM mode.

25.3.15 TSG3nSTR0 — TSG3n Status Register 0

This register controls the flags.

Access: This register can be read only in 8-bit units.

Address: <TSG3n_base> + 010_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|----------|----------|----------|---------|
| | — | — | — | — | TSG3nCUF | TSG3nSUF | TSG3nRSF | TSG3nTE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 25.21 TSG3nSTR0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. |
| 3 | TSG3nCUF | <p>Indicates the count direction of the 18-bit counter.</p> <p>0: The 18-bit counter is incremented. 1: The 18-bit counter is decremented.</p> <p>TSG3nCUF is valid only in HT-PWM mode. In other modes, it is invalid (TSG3nCUF = 0).</p> |
| 2 | TSG3nSUF | <p>Indicates the count direction of the 18-bit sub-counter.</p> <p>0: The 18-bit sub-counter is incremented. 1: The 18-bit sub-counter is decremented.</p> <ul style="list-style-type: none"> TSG3nSUF detects counting of the 18-bit sub-counter from 0000_H to (TSG3nCMP0E value - 0002_H) as up-counting, and counting from the TSG3nCMP0E value to 0002_H as down-counting. This bit is valid only in HT-PWM mode. |
| 1 | TSG3nRSF | <p>Indicates whether there is a reload request.</p> <p>0: No reload request or reload has completed. 1: There is a reload request.</p> <ul style="list-style-type: none"> This bit is valid only in TSG3nRMC = 0. This bit indicates that the data to be transferred next is held. This bit is set to 1 by writing to registers to be reloaded, and cleared to 0 when reload has completed. When TSG3nRMC is changed from 0 to 1 in HT-PWM mode, TSG3nRSF is cleared to 0. <p>For registers to be reloaded, see Section 25.3.1, List of Registers.</p> |
| 0 | TSG3nTE | <p>Indicates the TSG3n operation status.</p> <p>0: TSG3n is stopped. 1: TSG3n is operating.</p> <p>This bit is set when TSG3nTRG0.TSG3nTS = 1, and cleared when TSG3nTRG1.TSG3nTT = 1.</p> |

25.3.16 TSG3nSTR1 — TSG3n Status Register 1

This register controls the flags.

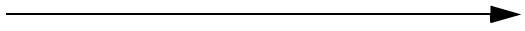
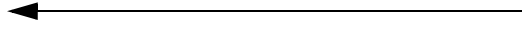
Access: This register can be read only in 8-bit units.

Address: <TSG3n_base> + 014_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|----------|---------------|---|---|
| | — | — | — | — | TSG3nTSF | TSG3nOPF[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 25.22 TSG3nSTR1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 7 to 4 | Reserved | When read, the value after reset is read. |
| 3 | TSG3nTSF | <p>Indicates the pattern change order of TSG3nPTSI0 to TSG3nPTSI2.</p> <p>0: Indicates that patterns are input to TSG3nPTSI0 to TSG3nPTSI2 in the normal rotation pattern order</p> <p>1: Indicates that patterns are input to TSG3nPTSI0 to TSG3nPTSI2 in the reverse rotation pattern order.</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>Normal Rotation</p>  </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>Reverse Rotation</p>  </div> <p style="margin-top: 10px;">TSG3nPTSI2- TSG3nPTSI0 [1,0,1] [1,0,0] [1,1,0] [0,1,0] [0,1,1] [0,0,1]</p> <p>Normal or reverse rotation can be detected from the first change of TSG3nPTSI0 to TSG3nPTSI2 after TSG3nTRG0.TSG3nTS has been set to 1. For details, see Section 25.4.3.5 (b), Detection of Change in Input Pattern Order.</p> |
| 2 to 0 | TSG3nOPF [2:0] | Indicates the output pattern of the timer output pins (TSG3nO1 to TSG3nO6). |

25.3.17 TSG3nSTR2 — TSG3n Status Register 2

This register controls the flags.

Access: This register can be read only in 16-bit units.

Address: <TSG3n_base> + 018_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|---|
| | — | — | — | — | — | — | TSG3nTBF2 | TSG3nTBF1 | TSG3nTBF0 | TSG3nPPF | TSG3nPEF | TSG3nTDF | TSG3nNDF | TSG3nPRF | TSG3nPTF | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 25.23 TSG3nSTR2 Register Contents (1/3)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 15 to 10 | Reserved | When read, the value after reset is read. |
| 9 | TSG3nTBF2 | <p>Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSG3nCTL1.TSG3nTBA2 is 1.</p> <p>0: Positive phase (TSG3nO5) and inverse phase (TSG3nO6) are not active simultaneously.</p> <p>1: Positive phase (TSG3nO5) and inverse phase (TSG3nO6) are active simultaneously.</p> <ul style="list-style-type: none"> TSG3nTBF2 is set to 1 when the simultaneous active state of the positive phase (TSG3nO5) and inverse phase (TSG3nO6) is detected, and an error interrupt (INTTSG3nIER) is generated. TSG3nTBF2 can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start) or by writing 1 to TSG3nSTC.TSG3nTBR2. The simultaneous active state is not detected when TSG3nTBA2 = 0. |
| 8 | TSG3nTBF1 | <p>Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSG3nCTL1.TSG3nTBA1 is 1.</p> <p>0: Positive phase (TSG3nO3) and inverse phase (TSG3nO4) are not active simultaneously.</p> <p>1: Positive phase (TSG3nO3) and inverse phase (TSG3nO4) are active simultaneously.</p> <ul style="list-style-type: none"> TSG3nTBF1 is set to 1 when the simultaneous active state of the positive phase (TSG3nO3) and inverse phase (TSG3nO4) is detected, and an error interrupt (INTTSG3nIER) is generated. TSG3nTBF1 can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start) or by writing 1 to TSG3nSTC.TSG3nTBR1. The simultaneous active state is not detected when TSG3nTBA1 = 0. |
| 7 | TSG3nTBF0 | <p>Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSG3nCTL1.TSG3nTBA0 is 1.</p> <p>0: Positive phase (TSG3nO1) and inverse phase (TSG3nO2) are not active simultaneously.</p> <p>1: Positive phase (TSG3nO1) and inverse phase (TSG3nO2) are active simultaneously.</p> <ul style="list-style-type: none"> TSG3nTBF0 is set to 1 when the simultaneous active state of the positive phase (TSG3nO1) and inverse phase (TSG3nO2) is detected, and an error interrupt (INTTSG3nIER) is generated. TSG3nTBF0 can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start) or by writing 1 to TSG3nSTC.TSG3nTBR0. The simultaneous active state is not detected when TSG3nTBA0 = 0. |

Table 25.23 TSG3nSTR2 Register Contents (2/3)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 6 | TSG3nPPF | <p>Indicates detection of the difference between the input patterns (TSG3nPTSI0 to TSG3nPTSI2) and the output patterns (TSG3nO1 to TSG3nO6) after they are compared.</p> <p>0: No phase difference detected between the input patterns (TSG3nPTSI0 to TSG3nPTSI2) and the output patterns (TSG3nO1 to TSG3nO6).</p> <p>1: A phase difference detected between the input patterns (TSG3nPTSI0 to TSG3nPTSI2) and the output patterns (TSG3nO1 to TSG3nO6).</p> <ul style="list-style-type: none"> TSG3nPPF is set to 1 when a difference between input and output patterns is detected, and a warning interrupt (INTTSG3nIWN) is generated. This bit can be cleared either by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), setting TSG3nTRG0.TSG3nTS to 1 (timer start), input to TSG3nTSST (timer restart), or writing 1 to TSG3nSTC.TSG3nPPR. |
| 5 | TSG3nPEF | <p>Indicates whether an abnormal input (000_B or 111_B) is input to TSG3nPTSI0 to TSG3nPTSI2) is detected.</p> <p>0: No abnormal input (000_B or 111_B) to TSG3nPTSI0 to TSG3nPTSI2 detected.</p> <p>1: Abnormal input (000_B or 111_B) to TSG3nPTSI0 to TSG3nPTSI2 detected.</p> <p>TSG3nPEF is set to 1 when an input of 000_B or 111_B to TSG3nPTSI0 to TSG3nPTSI2 is detected, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPEF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nPER.</p> <p>TSG3nPEF is valid when TSG3nCTL1.TSG3nPEC = 1.</p> |
| 4 | TSG3nTDF | <p>Indicates whether simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers is detected.</p> <p>0: Simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers not detected.</p> <p>1: Simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers detected.</p> <p>TSG3nTDF is set to 1 when simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers is detected, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nTDF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nTDR.</p> <p>TSG3nTDF is valid when TSG3nCTL1.TSG3nTDC = 1.</p> |
| 3 | TSG3nNDF | <p>Indicates whether noise on TSG3nPTSI0 to TSG3nPTSI2 is detected.</p> <p>0: Noise on TSG3nPTSI0 to TSG3nPTSI2 due to simultaneous change of two or more pins not detected.</p> <p>1: Noise on TSG3nPTSI0 to TSG3nPTSI2 due to simultaneous change of two or more pins detected.</p> <p>TSG3nNDF is set to 1 when simultaneous change of two or more pins in TSG3nPTSI0 to TSG3nPTSI2 is detected, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nNDF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nNDR.</p> <p>TSG3nNDF is valid when TSG3nCTL1.TSG3nNDC = 1.</p> |

Table 25.23 TSG3nSTR2 Register Contents (3/3)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 2 | TSG3nPRF | <p>Indicates whether reversal of the TSG3nPTSI0 to TSG3nPTSI2 input order is detected.</p> <p>0: The reversal of the TSG3nPTSI0 to TSG3nPTSI2 input order not detected. 1: The reversal of the TSG3nPTSI0 to TSG3nPTSI2 input order detected.</p> <p>TSG3nPRF is set to 1 when TSG3nSTR1.TSG3nTSF changes, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPRF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (timer operation start), by input to TSG3nTSST(timer restart), or by writing 1 to TSG3nSTC.TSG3nPRR. Detection is possible from the second TSG3nPTSI0 to TSG3nPTSI2 change timing after setting TSG3nTRG0.TSG3nTS = 1. TSG3nPRF is valid when TSG3nCTL1.TSG3nPRC = 1.</p> |
| 1 | TSG3nPTF | <p>Indicates whether an abnormal toggle of TSG3nPTSI0 to TSG3nPTSI2 is detected.</p> <p>0: No abnormal toggle of TSG3nPTSI0 to TSG3nPTSI2 detected. 1: An abnormal toggle of TSG3nPTSI0 to TSG3nPTSI2 detected.</p> <p>TSG3nPTF is set to 1 when TSG3nPTSI0 to TSG3nPTSI2 (TSG3nPTE signal toggle) are changed three times or more during TSG3nOPCI0 trigger or TSG3nPTSI0 to TSG3nPTSI2 (TSG3nPTE signal toggle) are changed three times or more during TSG3nOPCI1 trigger, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPTF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nPTR. TSG3nPTF is valid when TSG3nCTL1.TSG3nPTE[1:0] = 10_B or 11_B.</p> |
| 0 | Reserved | When read, the value after reset is read. |

25.3.18 TSG3nSTC — TSG3n Status Clear Trigger Register

This register controls the flags.

Access: This register can be written only in 16-bit units.

Address: <TSG3n_base> + 01C_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|---|
| | — | — | — | — | — | — | TSG3nTBR2 | TSG3nTBR1 | TSG3nTBR0 | TSG3nPPR | TSG3nPER | TSG3nTDR | TSG3nNDR | TSG3nPRR | TSG3nPTR | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | W | W | W | W | W | W | W | W | W | R |

Table 25.24 TSG3nSTC Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 15 to 10 | Reserved | When writing, write the value after reset. |
| 9 | TSG3nTBR2 | This is a trigger bit that clears TSG3nSTR2.TSG3nTBF2. 0: Does not clear TSG3nTBF2. 1: Clears TSG3nTBF2. When TSG3nTBR2 writing and TSG3nSTR2.TSG3nTBF2 setting occur simultaneously, TSG3nSTR2.TSG3nTBF2 setting has a priority, and the flag is not cleared. |
| 8 | TSG3nTBR1 | This is a trigger bit that clears TSG3nSTR2.TSG3nTBF1. 0: Does not clear TSG3nTBF1. 1: Clears TSG3nTBF1. When TSG3nTBR1 writing and TSG3nSTR2.TSG3nTBF1 setting occur simultaneously, TSG3nSTR2.TSG3nTBF1 setting has a priority, and the flag is not cleared. |
| 7 | TSG3nTBR0 | This is a trigger bit that clears TSG3nSTR2.TSG3nTBF0. 0: Does not clear TSG3nTBF0. 1: Clears TSG3nTBF0. When TSG3nTBR0 writing and TSG3nSTR2.TSG3nTBF0 setting occur simultaneously, TSG3nSTR2.TSG3nTBF0 setting has a priority, and the flag is not cleared. |
| 6 | TSG3nPPR | This is a trigger bit that clears TSG3nSTR2.TSG3nPPF. 0: Does not clear TSG3nPPF. 1: Clears TSG3nPPF. When TSG3nPPR writing and TSG3nSTR2.TSG3nPPF setting occur simultaneously, TSG3nSTR2.TSG3nPPF setting has a priority, and the flag is not cleared. |
| 5 | TSG3nPER | This is a trigger bit that clears TSG3nSTR2.TSG3nPEF. 0: Does not clear TSG3nPEF. 1: Clears TSG3nPEF. When TSG3nPER writing and TSG3nSTR2.TSG3nPEF setting occur simultaneously, TSG3nSTR2.TSG3nPEF setting has a priority, and the flag is not cleared. |
| 4 | TSG3nTDR | This is a trigger bit that clears TSG3nSTR2.TSG3nTDF. 0: Does not clear TSG3nTDF. 1: Clears TSG3nTDF. When TSG3nTDR writing and TSG3nSTR2.TSG3nTDF setting occur simultaneously, TSG3nSTR2.TSG3nTDF setting has a priority, and the flag is not cleared. |

Table 25.24 TSG3nSTC Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 3 | TSG3nNDR | <p>This is a trigger bit that clears TSG3nSTR2.TSG3nNDF. 0: Does not clear TSG3nNDF. 1: Clears TSG3nNDF.</p> <p>When TSG3nNDR writing and TSG3nSTR2.TSG3nNDF setting occur simultaneously, TSG3nSTR2.TSG3nNDF setting has a priority, and the flag is not cleared.</p> |
| 2 | TSG3nPRR | <p>This is a trigger bit that clears TSG3nSTR2.TSG3nPRF. 0: Does not clear TSG3nPRF. 1: Clears TSG3nPRF.</p> <p>When TSG3nPRR writing and TSG3nSTR2.TSG3nPRF setting occur simultaneously, TSG3nSTR2.TSG3nPRF setting has a priority, and the flag is not cleared.</p> |
| 1 | TSG3nPTR | <p>This is a trigger bit that clears TSG3nSTR2.TSG3nPTF. 0: Does not clear TSG3nPTF. 1: Clears TSG3nPTF.</p> <p>When TSG3nPTR writing and TSG3nSTR2.TSG3nPTF setting occur simultaneously, TSG3nSTR2.TSG3nPTF setting has a priority, and the flag is not cleared.</p> |
| 0 | Reserved | When writing, write the value after reset. |

25.3.19 TSG3nOPT0 — TSG3n Option Register 0

This register sets the optional functions.

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> + 020_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|----------|----------|----------|----------|----------|----------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | TSG3nSOC | TSG3nSTE | TSG3nPOT | TSG3nPSS | TSG3nIDC | TSG3nPSC | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 25.25 TSG3nOPT0 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | TSG3nSOC | Enables or disables control of the timer output (TSG3nO1 to TSG3nO6 pins) by software. 0: Disables control by software. 1: Enables control by software. When TSG3nSOC is set to 1, timer output is switched to the software control/trigger control output pattern specified by TSG3nSPC2 to TSG3nSPC0. The dead time is secured by the dead time counter. |
| 5 | TSG3nSTE | Enables or disables control by the pattern output trigger. 0: Disables the TSG3nPTSI0 to TSG3nPTSI2 and TSG3nOPCI0, and TSG3nOPCI1 inputs. 1: Enables TSG3nPTSI0 to TSG3nPTSI2 and TSG3nOPCI0, and TSG3nOPCI1 inputs. <ul style="list-style-type: none"> The pattern output trigger is selected by TSG3nPOT. TSG3nSTE is valid in 120-DC mode and when software output control function is enabled. |
| 4 | TSG3nPOT | Selects the pattern output trigger. 0: Switches the output pattern by the external pattern input pins (TSG3nPTSI0 to TSG3nPTSI2) (pattern switch method). 1: Switches the output pattern by the rising edge of the TSG3nOPCI0 and TSG3nOPCI1 (trigger switch method). |
| 3 | TSG3nPSS | Selects the pattern output order switch factor. 0: The pattern output order is not switched by TSG3nPSC. 1: The pattern output order is switched by TSG3nPSC. |
| 2 | TSG3nIDC | Determines the output pattern from the TSG3nO1 to TSG3nO6 pins in combination with the TSG3nIDC and TSG3nSTR1.TSG3nTSF and TSG3nPSC signals. For the timer output order and patterns to be output, see Figure 25.80 to Figure 25.83 , Example of Operation in 120-DC Mode, in Section 25.4.7.4 (5), Operation in 120-DC Mode . |

Table 25.25 TSG3nOPT0 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 1 | TSG3nPSC | <p>Selects the pattern output order when the semi-automatic cruise function is enabled.</p> <p>0: Switches the timer output (TSG3nO1 to TSG3nO6) in the normal rotation. 1: Switches the timer output (TSG3nO1 to TSG3nO6) in the reverse rotation.</p> <ul style="list-style-type: none"> • TSG3nPSC specifies the timer output pattern order assuming the output pattern specified by TSG3nSPC2 to TSG3nSPC0 as the initial pattern. TSG3nPSC is valid when TSG3nPOT = 1 and TSG3nPSS = 1. • It is recommended to rewrite TSG3nPSC when TSG3nSTR0.TSG3nTE = 0 or TSG3nPOT = 0. If TSG3nPSC is rewritten when TSG3nPOT = 1, unexpected timer output pattern might be caused. • If the signal input to TSG3nPTSI0 to TSG3nPTSI2 changes with TSG3n operation being stopped (TSG3nSTR0.TSG3nTE = 0), the TSG3nTRG0.TSG3nTS bit should be set to 1 after matching the input signal change logic with the TSG3nPSC order. • For output order in normal or reverse rotation, see Section 25.4.7.4, 120-DC Mode. Here, normal rotation and reverse rotation refer to the change of output, and they are different from normal rotation and reverse rotation of a motor. |
| 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

25.3.20 TSG3nOPT1 — TSG3n Option Register 1

This register sets the optional functions.

Access: This register can be read/written in 8-bit units.

Address: <TSG3n_base> +024_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---------------|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | TSG3nSPC[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 25.26 TSG3nOPT1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 7 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | TSG3nSPC [2:0] | Specifies the timer output pattern when software output function is enabled and in 120-DC mode. For the output pattern, see Section 25.4.7.8, Software Output Control Function , and Section 25.4.7.4, 120-DC Mode . |

25.3.21 TSG3nTRG0 — TSG3n Trigger Register 0

This register controls the start of the timer.

Access: This register can be written only in 8-bit units.

Address: <TSG3n_base> + 030_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | TSG3nTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 25.27 TSG3nTRG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset (or the fixed value). |
| 0 | TSG3nTS | This bit is a trigger bit that controls the start of the timer. 0: The timer is not started. 1: The timer is started (restarted if TSG3nSTR0.TSG3nTE = 1). When restarted, the 18-bit counter is initialized. This bit is always read as 0. |

25.3.22 TSG3nTRG1 — TSG3n Trigger Register 1

This register controls the stop of the timer.

Access: This register can be written only in 8-bit units.

Address: <TSG3n_base> +034_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | TSG3nTT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 25.28 TSG3nTRG1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TSG3nTT | This is a trigger bit that controls the stop of the timer. 0: The timer is not stopped. 1: The timer is stopped (TSG3nSTR0.TSG3nTE = 0). This bit is always read as 0. |

25.3.23 TSG3nTRG2 — TSG3n Trigger Register 2

TSG3nTRG2 is a trigger bit to reflect the PWM duty setting to TSG3nO1-6 in anytime rewrite mode of HT-PWM mode.

This register can be set to 1 only in HT-PWM mode and when anytime rewrite mode is selected (TSG3nRMC = 1). Do not rewrite this register in other modes (PWM mode, SP-PMW mode, 120-DC mode, HSP-PWM mode) or when reload mode is selected (TSG3nRMC = 0)

Access: This register can be written only in 8-bit units.

Address: <TSG3n_base> + 038_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|----------|
| | — | — | — | — | — | — | — | TSG3nIMT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 25.29 TSG3nTRG2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TSG3nIMT | Anytime Rewrite Trigger 0: Disabled 1: Changes in duty settings for U, V, and W phases are reflected to timer output in HT-PWM mode and when anytime rewrite mode is selected. |

25.3.24 TSG3nCNT — TSG3n Counter Read Buffer Register

This register can access the 16 lower bits of the 18-bit register TSG3nCnTE.

For the operation of this register, see **Section 25.3.25, TSG3nCnTE — TSG3n Bit Extended Counter Read Buffer Register.**

Access: This register can be read only in 16-bit units.

Address: <TSG3n_base> + 028_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | 16-bit counter | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

25.3.25 TSG3nCnTE — TSG3n Bit Extended Counter Read Buffer Register

The counter values can be read from this register. This register mirrors the contents of TSG3nCnT from which the 16 lower bits of this register can be accessed.

Access: This register can be read only in 32-bit units.

Address: <TSG3n_base> + 1A0_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------------------|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nCnTE (18-bit counter) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | TSG3nCnTE (18-bit counter) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

18-bit counter

This register is a timer read buffer register from which the 18-bit counter value can be read. In HT-PWM mode, the 18-bit counter provides the triangular waveform control in which the counter value is incremented and decremented by 2. Bit 0 is always read as 0.

In other modes, the 18-bit counter provides the sawtooth waveform control in which the counter value is incremented by 1.

Table 25.30 TSG3nCnTE Register Count Value

| Operating mode | At the Beginning | Minimum Value | Maximum value |
|----------------|--------------------|--------------------|-------------------------------------|
| HT-PWM mode | TSG3nDTC0 | TSG3nDTC0 | TSG3nDTC0+TSG3nCnMP0E ^{*1} |
| Other modes | 00000 _H | 00000 _H | TSG3nCnMP0E |

Note 1. Set the value as TSG3nDTC0+TSG3nCnMP0E < 3FFFF_H.

25.3.26 TSG3nSBC — TSG3n Sub-Counter Read Buffer Register

This register can access the 16 lower bits of the 18-bit register TSG3nSBCE.

For the operation of this register, see **Section 25.3.27, TSG3nSBCE — TSG3n Bit Extended Sub-Counter Read Buffer Register**.

Access: This register can be read only in 16-bit units.

Address: <TSG3n_base> + 02C_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | 16-bit sub-counter | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

25.3.27 TSG3nSBCE — TSG3n Bit Extended Sub-Counter Read Buffer Register

The sub-counter values can be read from this register. This register mirrors the contents of TSG3nSBC from which the 16 lower bits of this register can be accessed.

Access: This register can be read only in 32-bit units.

Address: <TSG3n_base> + 1A4_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------------------------|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nSBCE (18-bit sub-counter) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nSBCE (18-bit sub-counter) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

18-bit counter

This register is a timer read buffer register from which the 18-bit sub-counter value can be read. In HT-PWM mode, the 18-bit counter provides the triangular waveform control in which the counter value is incremented and decremented by 2. Bit 0 is always read as 0. (Available only in HT-PWM mode.).

Table 25.31 TSG3nSBCE Register Count Value

| Operating mode | At the Beginning | Minimum Value | Maximum Value |
|----------------|--------------------|--------------------|--|
| HT-PWM mode | TSG3nDTC0 | 00000 _H | TSG3nDTC0 + TSG3nDTC1 + TSG3nCMP0E ^{*1} |
| Other modes | 00000 _H | 00000 _H | 00000 _H |

Note 1. Set the value as TSG3nDTC0 + TSG3nDTC1 + TSG3nCMP0E < 3FFFF_H.

25.3.28 TSG3nCMP0 — TSG3n Compare Register 0

This register can access the 16 lower bits of the 18-bit register TSG3nCMP0E.

For the operation of this register, see **Section 25.3.29, TSG3nCMP0E — TSG3n Bit Extended Compare Register 0**.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 058_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | 16-bit compare register | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.29 TSG3nCMP0E — TSG3n Bit Extended Compare Register 0

This register is an 18-bit compare register that specifies the PWM period in all modes. This register mirrors the contents of TSG3nCMP0 from which the 16 lower bits of this register can be accessed.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 14C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nCMP0E (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nCMP0E (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 25.32 TSG3nCMP0E Register Setting

| Operating mode | At the Beginning | Minimum Value | Maximum Value |
|----------------|--------------------------|--------------------------------------|---|
| HT-PWM mode | TSG3nCMP0E ^{*1} | 00002 _H | 3FFFE _H |
| Other modes | TSG3nCMP0E + 1 | 1 (TSG3nCMP0E = 00000 _H) | 40000 _H (TSG3nCMP0E = 3FFFF _H) |

Note 1. In HT-PWM mode, the lowest bit is ignored.

25.3.30 TSG3nCMP1W — TSG3n Compare Register 1, 2

This register can access the 16 lower bits of the 18-bit register TSG3nCMP1E and TSG3nCMP2E.

For the operation of this register, see **Section 25.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 040_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nCMP2 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSG3nCMP1 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.31 TSG3nCMP3W — TSG3n Compare Register 3, 4

This register can access the 16 lower bits of the 18-bit register TSG3nCMP3E and TSG3nCMP4E.

For the operation of this register, see **Section 25.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 04C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nCMP4 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSG3nCMP3 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.32 TSG3nCMP5W — TSG3n Compare Register 5, 6

This register can access the 16 lower bits of the 18-bit register TSG3nCMP5E and TSG3nCMP6E.

For the operation of this register, see **Section 25.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 044_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nCMP6 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSG3nCMP5 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.33 TSG3nCMP7W — TSG3n Compare Registers 7, 8

This register can access the 16 lower bits of the 18-bit register TSG3nCMP7E and TSG3nCMP8E.

For the operation of this register, see **Section 25.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 050_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nCMP8 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSG3nCMP7 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.34 TSG3nCMP9W — TSG3n Compare Registers 9, 10

This register can access the 16 lower bits of the 18-bit register TSG3nCMP9E and TSG3nCMP10E.

For the operation of this register, see **Section 25.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 048_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TSG3nCMP10 (16-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nCMP9 (16-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.35 TSG3nCMP11W — TSG3n Compare Registers 11, 12

This register can access the 16 lower bits of the 18-bit register TSG3nCMP11E and TSG3nCMP12E.

For the operation of this register, see **Section 25.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 054_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TSG3nCMP12 (16-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nCMP11 (16-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.36 TSG3nCMP1 to TSG3nCMP12 — TSG3n Compare Registers 1 to 12

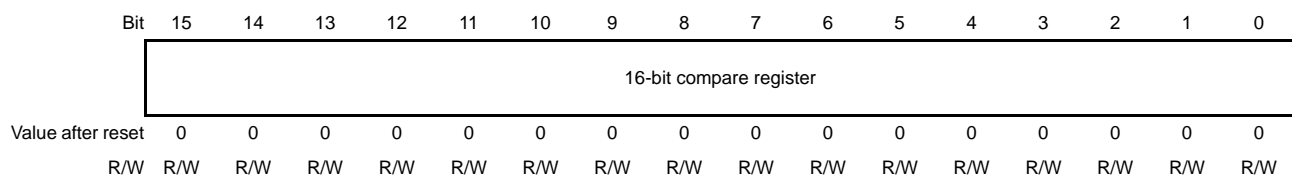
These registers can access the 16 lower bits of the 18-bit registers TSG3nCMP1E-12E.

For the operation of these registers, see **Section 25.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12.**

Access: This register can be read/written in 16-bit units.

- Address:** TSG3nCMP1 <TSG3n_base> + 080_H
 TSG3nCMP2 <TSG3n_base> + 084_H
 TSG3nCMP3 <TSG3n_base> + 098_H
 TSG3nCMP4 <TSG3n_base> + 09C_H
 TSG3nCMP5 <TSG3n_base> + 088_H
 TSG3nCMP6 <TSG3n_base> + 08C_H
 TSG3nCMP7 <TSG3n_base> + 0A0_H
 TSG3nCMP8 <TSG3n_base> + 0A4_H
 TSG3nCMP9 <TSG3n_base> + 090_H
 TSG3nCMP10 <TSG3n_base> + 094_H
 TSG3nCMP11 <TSG3n_base> + 0A8_H
 TSG3nCMP12 <TSG3n_base> + 0AC_H

Value after reset: 0000_H



25.3.37 TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12

The compare value is set by these registers. These registers mirror the contents of TSG3nCMP1-12, TSG3nCMP1W, 3W, 5W, 7W, 9W, and 11W from which the 16 lower bits of these registers can be accessed.

Access: This register can be read/written in 32-bit units.

Address: TSG3nCMP1E <TSG3n_base> + 17C_H
 TSG3nCMP2E <TSG3n_base> + 178_H
 TSG3nCMP3E <TSG3n_base> + 164_H
 TSG3nCMP4E <TSG3n_base> + 160_H
 TSG3nCMP5E <TSG3n_base> + 174_H
 TSG3nCMP6E <TSG3n_base> + 170_H
 TSG3nCMP7E <TSG3n_base> + 15C_H
 TSG3nCMP8E <TSG3n_base> + 158_H
 TSG3nCMP9E <TSG3n_base> + 16C_H
 TSG3nCMP10E <TSG3n_base> + 168_H
 TSG3nCMP11E <TSG3n_base> + 154_H
 TSG3nCMP12E <TSG3n_base> + 150_H

Value after reset: 0000 0000_H

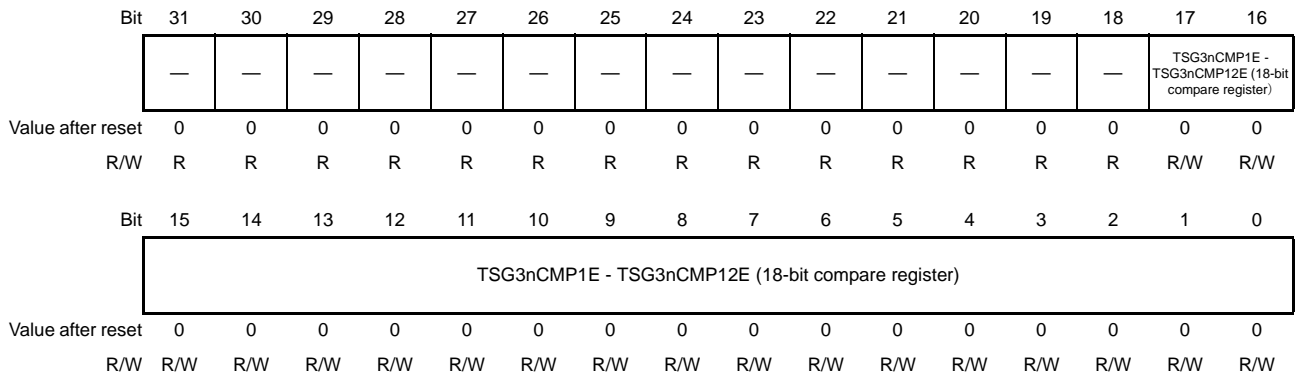


Table 25.33 TSG3nCMP1E-TSG3nCMP12E Register Setting (1/2)

| Register | PWM Mode | HT-PWM Mode | SP-PWM Mode | 120-DC Mode | HSP-PWM Mode |
|-------------|----------------------|--|-------------|---|----------------------|
| TSG3nCMP1E | TSG3nO1 clear timing | TSG3nO1 clear timing /TSG3nO2 set timing | | Duty when TSG3nO1, 3, or 5 output pattern is selected by TSG3nPAT0. | TSG3nO1 clear timing |
| TSG3nCMP2E | TSG3nO1 set timing | TSG3nO1 set timing/TSG3nO2 clear timing | | | TSG3nO1 set timing |
| TSG3nCMP3E | TSG3nO2 clear timing | — | | Duty when TSG3nO2, 4, or 6 output pattern is selected by TSG3nPAT1. | TSG3nO2 clear timing |
| TSG3nCMP4E | TSG3nO2 set timing | — | | | TSG3nO2 set timing |
| TSG3nCMP5E | TSG3nO3 clear timing | TSG3nO3 clear timing/ TSG3nO4 set timing | | Duty when TSG3nO1, 3, or 5 output pattern is selected by TSG3nPAT0. | TSG3nO3 clear timing |
| TSG3nCMP6E | TSG3nO3 set timing | TSG3nO3 set timing/TSG3nO4 clear timing | | | TSG3nO3 set timing |
| TSG3nCMP7E | TSG3nO4 clear timing | — | | Duty when TSG3nO2, 4, or 6 output pattern is selected by TSG3nPAT1. | TSG3nO4 clear timing |
| TSG3nCMP8E | TSG3nO4 set timing | — | | | TSG3nO4 set timing |
| TSG3nCMP9E | TSG3nO5 clear timing | TSG3nO5 clear timing/ TSG3nO6 set timing | | Duty when TSG3nO1, 3, or 5 output pattern is selected by TSG3nPAT0. | TSG3nO5 clear timing |
| TSG3nCMP10E | TSG3nO5 set timing | TSG3nO5 set timing/TSG3nO6 clear timing | | | TSG3nO5 set timing |

Table 25.33 TSG3nCMP1E-TSG3nCMP12E Register Setting (2/2)

| Register | PWM Mode | HT-PWM Mode | SP-PWM Mode | 120-DC Mode | HSP-PWM Mode |
|-------------|----------------------|-------------|-------------|---|----------------------|
| TSG3nCMP11E | TSG3nO6 clear timing | — | | Duty when TSG3nO2, 4, or 6 output pattern is selected by TSG3nPAT1. | TSG3nO6 clear timing |
| TSG3nCMP12E | TSG3nO6 set timing | — | | | TSG3nO6 set timing |

NOTE

The dead time function is used in all operating modes.

In HT-PWM mode, the compare match signal is generated when a compare match occurs not only with the value of the TSG3nCNTE register but also with the value of the TSG3nSBCE register.

In 120-DC mode, the output from TSG3nO1-6 is controlled by the TSG3nCMPmE, TSG3nPAT0, and TSG3nPAT1 registers.

25.3.38 TSG3nDCMP0W — TSG3n Diagnostic Output Compare Register 0, 1

This register can access the 16 lower bits of the 18-bit register TSG3nDCMP0E and TSG3nDCMP1E.

For the operation of this register, see **Section 25.3.40, TSG3nDCMP0E to 2E — TSG3n Bit Extended Diagnostic Output Compare Register 0 to 2.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 05C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nDCMP1(16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSG3nDCMP0(16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.39 TSG3nDCMP2 — TSG3n Diagnostic Output Compare Register 2

This register can access the 16 lower bits of the 18-bit register TSG3nDCMP2E.

For the operation of this register, see **Section 25.3.40, TSG3nDCMP0E to 2E — TSG3n Bit Extended Diagnostic Output Compare Register 0 to 2.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 060_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nDCMP2 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSG3nDCMP2 (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.40 TSG3nDCMP0E to 2E — TSG3n Bit Extended Diagnostic Output Compare Register 0 to 2

The compare value is set by these registers. These registers mirror the contents of TSG3nDCMP0W and TSG3nDCMP2 from which the 16 lower bits of these registers can be accessed.

Access: This register can be read/written in 32-bit units.

Address: TSG3nDCMP0E <TSG3n_base> + 148_H
 TSG3nDCMP1E <TSG3n_base> + 144_H
 TSG3nDCMP2E <TSG3n_base> + 140_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nDCMP0E - TSG3nDCMP2E (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nDCMP0E - TSG3nDCMP2E (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

These registers control the diagnostic output timing or AD conversion trigger timing in all modes. A pulse is generated at the match timing of the 18-bit counter value with this register.

25.3.41 TSG3nPAT0W — TSG3n Pattern Register 0

This register specifies the output pattern.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 064_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | | |
|-------------------|-------|----|-------|-----|-----|-------|-----|-----|-------|-----|-----|-------|-----|-----|-------|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PAT5T | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | |
| Bit | 15 | | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PAT5T | | PAT4T | | | PAT3T | | | PAT2T | | | PAT1T | | | PAT0T | | |
| Value after reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Output Pattern

This register controls UT/VT/WT output in 120-DC mode.

Table 25.34 TSG3nPAT0W Register Setting value and Output Control

| PATmT Value | Output Control |
|-------------|-------------------------------|
| 000 | Fixed Low level |
| 001 | PWM output set by TSG3nCMP1E |
| 010 | PWM output set by TSG3nCMP2E |
| 011 | PWM output set by TSG3nCMP5E |
| 100 | PWM output set by TSG3nCMP6E |
| 101 | PWM output set by TSG3nCMP9E |
| 110 | PWM output set by TSG3nCMP10E |
| 111 | Fixed High level |

Note: m = 0, 1, 2, 3, 4, 5

25.3.42 TSG3nPAT1W — TSG3n Pattern Register 1

This register specifies the output pattern.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 068_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-----|-------|-----|-----|-------|-----|-----|-------|-----|-----|-------|-----|-----|-------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PAT5B | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PAT5B | | PAT4B | | | PAT3B | | | PAT2B | | | PAT1B | | | PAT0B | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Output Pattern

This register controls UB/VB/WB output in 120-DC mode.

Table 25.35 TSG3nPAT1W Register Setting value and Output Control

| PATmB Value | Output Control |
|-------------|-------------------------------|
| 000 | Fixed Low level |
| 001 | PWM output set by TSG3nCMP3E |
| 010 | PWM output set by TSG3nCMP4E |
| 011 | PWM output set by TSG3nCMP7E |
| 100 | PWM output set by TSG3nCMP8E |
| 101 | PWM output set by TSG3nCMP11E |
| 110 | PWM output set by TSG3nCMP12E |
| 111 | Fixed High level |

Note: m = 0, 1, 2, 3, 4, 5

25.3.43 TSG3nDTC0W — TSG3n Dead Time Control Register 0

This register sets the dead time value (the period from inverse phase inactivation to positive phase activation).

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 06C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------------------------------|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | Write Protection Code Check | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nDTC0(10-bit dead time compare) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

To rewrite TSG3nDTC0W[0:9], set bit 14 to bit 0 and TSG3nDTCM to 0 in TSG3nDTPR, and rewrite the TSG3nDTC0W. At this time, when the rewritten value of TSG3nDTC0W[30:16] and the TSG3nDTPR value match, TSG3nDTC0W is rewritten.

During timer operation (TSG3nSTR0.TSG3nTE = 1), rewriting should be performed in reload mode (TSG3nCTL3.TSG3nRMC = 0).

25.3.44 TSG3nDTC1W — TSG3n Dead Time Control Register 1

This register sets the dead time (the period from positive phase inactivation to inverse phase activation).

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 070_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------------------------------|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | Write Protection Code Check | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nDTC1(10-bit dead time compare) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

To rewrite TSG3nDTC1W[0:9], set bit 14 to bit 0 and TSG3nDTCM to 0 in TSG3nDTPR, and rewrite the TSG3nDTC1W. At this time, when the rewritten value of TSG3nDTC1W[30:16] and the TSG3nDTPR value match, TSG3nDTC1W is rewritten.

During timer operation (TSG3nSTR0.TSG3nTE = 1), rewriting should be performed in reload mode (TSG3nCTL3.TSG3nRMC = 0).

25.3.45 TSG3nCMPU — TSG3n HT-PWM U Phase Compare Register

This register can access the 16 lower bits of the 18-bit register TSG3nCMPUE.

For the operation of this register, see **Section 25.3.48, TSG3nCMPUE — TSG3n Bit Extended HT-PWM U Phase Compare Register.**

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 0B0_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nCMPU(16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.46 TSG3nCMPV — TSG3n HT-PWM V Phase Compare Register

This register can access the 16 lower bits of the 18-bit register TSG3nCMPVE.

For the operation of this register, see **Section 25.3.49, TSG3nCMPVE — TSG3n Bit Extended HT-PWM V Phase Compare Register.**

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 0B4_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nCMPV (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.47 TSG3nCMPW — TSG3n HT-PWM W Phase Compare Register

This register can access the 16 lower bits of the 18-bit register TSG3nCMPWE.

For the operation of this register, see **Section 25.3.50, TSG3nCMPWE — TSG3n Bit Extended HT-PWM W Phase Compare Register.**

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 0B8_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nCMPW (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.48 TSG3nCMPUE — TSG3n Bit Extended HT-PWM U Phase Compare Register

This register sets the compare value for U phase in HT-PWM. In addition to the functions of TSG3nCMP1E and TSG3nCMP2E, this register can access specific registers.

The data written to this register is stored in the TSG3nCMP1E and TSG3nCMP2E registers which allows a generation of symmetry triangular waveform of PWM by one write access (see **Figure 25.2**). When this register is read, the same value as TSG3nCMP1E is returned. This register mirrors the contents of TSG3nCMPU from which the 16 lower bits of this register can be accessed.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> +188_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | | | | | | TSG3nCMPUE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nCMPUE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

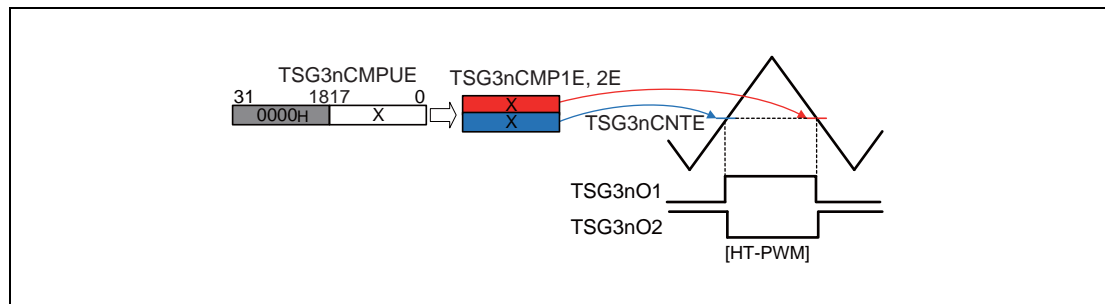


Figure 25.2 TSG3nCMPUE Register Accesses

25.3.49 TSG3nCMPVE — TSG3n Bit Extended HT-PWM V Phase Compare Register

This register sets the compare value for V phase in HT-PWM. In addition to the functions of TSG3nCMP5E and TSG3nCMP6E, this register can access specific registers.

The data written to this register is stored in the TSG3nCMP5E and TSG3nCMP6E registers which allows a generation of symmetry triangular waveform of PWM by one write access (see **Figure 25.3**). When this register is read, the same value as TSG3nCMP5E is returned. This register mirrors the contents of TSG3nCMPV from which the 16 lower bits of this register can be accessed.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> +184_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | | | | | | TSG3nCMPVE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nCMPVE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

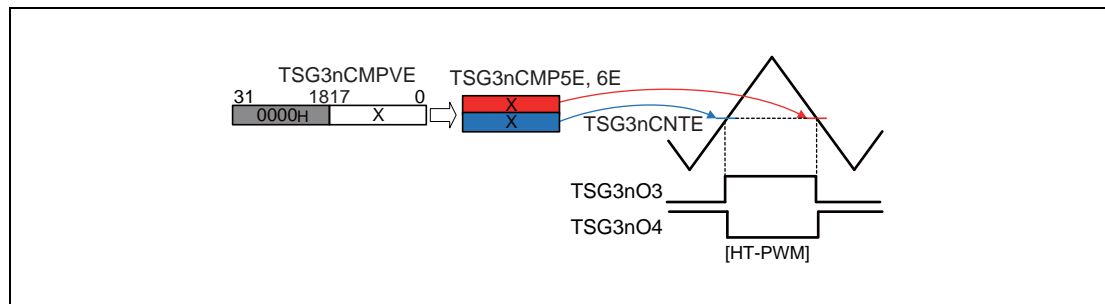


Figure 25.3 TSG3nCMPVE Register Accesses

25.3.50 TSG3nCMPWE — TSG3n Bit Extended HT-PWM W Phase Compare Register

This register sets the compare value for W phase in HT-PWM. In addition to the functions of TSG3nCMP9E and TSG3nCMP10E, this register can access specific registers.

The data written to this register is stored in the TSG3nCMP9E and TSG3nCMP10E registers which allows a generation of symmetry triangular waveform of PWM by one write access (see **Figure 25.4**). When this register is read, the same value as TSG3nCMP9E is returned. This register mirrors the contents of TSG3nCMPW from which the 16 lower bits of this register can be accessed.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> +180_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------------------------------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nCMPWE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nCMPWE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

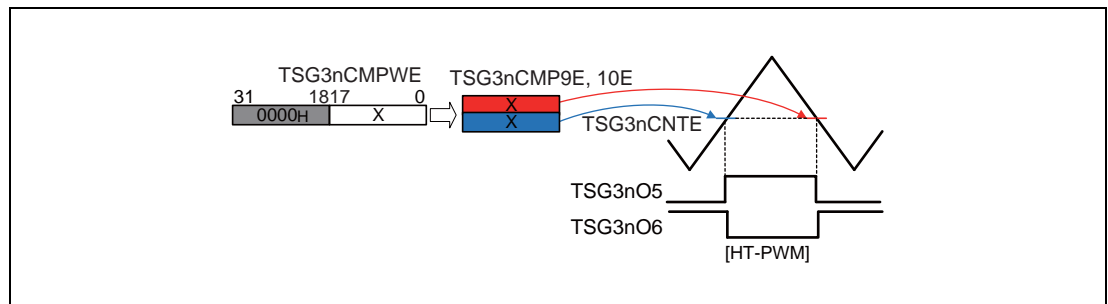


Figure 25.4 TSG3nCMPWE Register Accesses

25.3.51 TSG3nUPW — TSG3n SP-PWM U Phase Active Width Register

This register can access the 16 lower bits of the 18-bit register TSG3nUPWE.

For the operation of this register, see **Section 25.3.54, TSG3nUPWE — TSG3n Bit Extended SP-PWM U Phase Active Width Register.**

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 0BC_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nUPW (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.52 TSG3nVPW — TSG3n SP-PWM V Phase Active Width Register

This register can access the 16 lower bits of the 18-bit register TSG3nVPWE.

For the operation of this register, see **Section 25.3.55, TSG3nVPWE — TSG3n Bit Extended SP-PWM V Phase Active Width Register.**

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 0C0_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nVPW (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.53 TSG3nWPW — TSG3n SP-PWM W Phase Active Width Register

This register can access the 16 lower bits of the 18-bit register TSG3nWPWE.

For the operation of this register, see **Section 25.3.56, TSG3nWPWE — TSG3n Bit Extended SP-PWM W Phase Active Width Register.**

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 0C4_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TSG3nWPW (16-bit compare register) | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.54 TSG3nUPWE — TSG3n Bit Extended SP-PWM U Phase Active Width Register

This register sets the active width for U phase in SP-PWM mode. The sum of the TSG3nUPWE write data and the TSG3nCMP2E value is stored in TSG3nCMP1E (see **Figure 25.5**). When this register is read, the same value as TSG3nCMP1E is returned. This register mirrors the contents of TSG3nUPW from which the 16 lower bits of this register can be accessed.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 198_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------------------------------------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nUPWE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nUPWE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

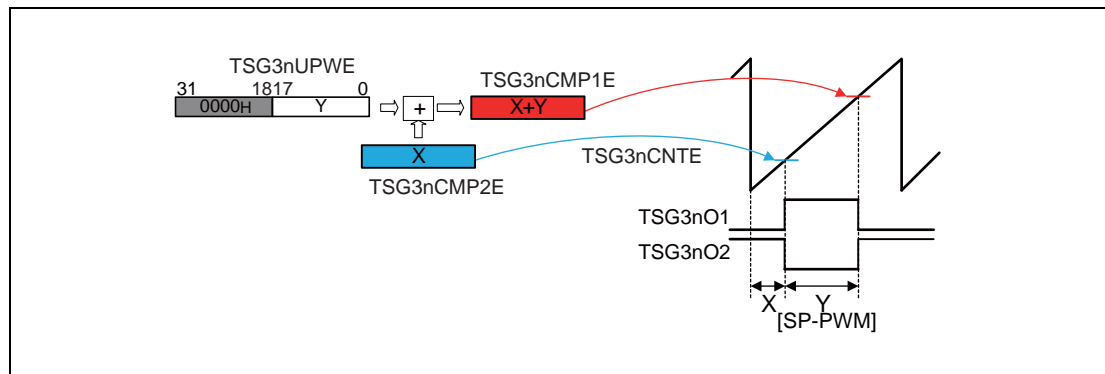


Figure 25.5 TSG3nUPWE Register Accesses

25.3.55 TSG3nVPWE — TSG3n Bit Extended SP-PWM V Phase Active Width Register

This register sets the active width for V phase in SP-PWM mode. The sum of the TSG3nVPWE write data and the TSG3nCMP6E value is stored in TSG3nCMP5E (see **Figure 25.6**). When this register is read, the same value as TSG3nCMP5E is returned. This register mirrors the contents of TSG3nVPW from which the 16 lower bits of this register can be accessed.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 194_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------------------------------------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nVPWE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nVPWE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

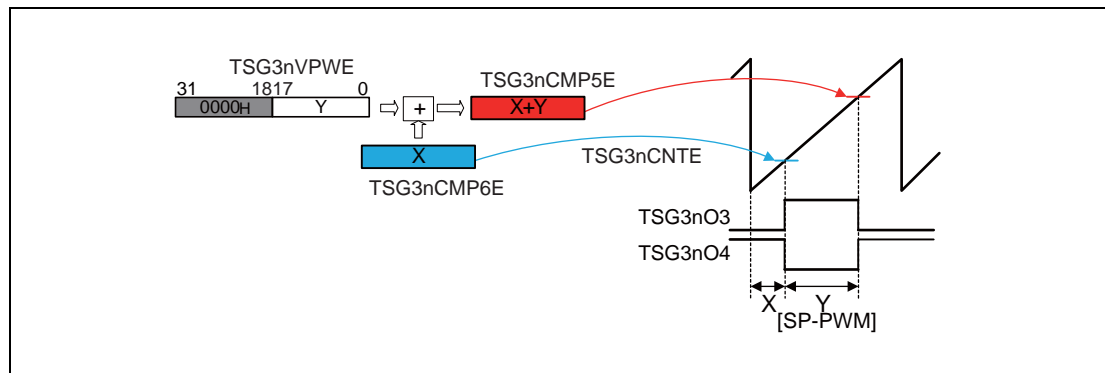


Figure 25.6 TSG3nVPWE Register Accesses

25.3.56 TSG3nWPWE — TSG3n Bit Extended SP-PWM W Phase Active Width Register

This register sets the active width for W phase in SP-PWM mode. The sum of the TSG3nWPWE write data and the TSG3nCMP10E value is stored in TSG3nCMP9E (see **Figure 25.7**). When this register is read, the same value as TSG3nCMP9E is returned. This register mirrors the contents of TSG3nWPWE from which the 16 lower bits of this register can be accessed.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 190_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------------------------------------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nWPWE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nWPWE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

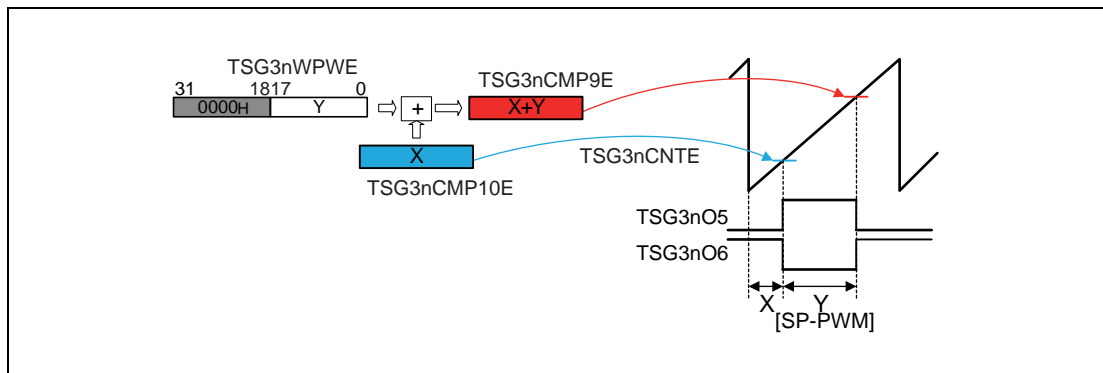


Figure 25.7 TSG3nWPWE Register Accesses

25.3.57 TSG3nHSPCMUE — TSG3n HSP-PWM Mode U Phase Compare Register

This register sets the PWM output width for U phase in HSP-PWM mode.

When a write access is made to this register, the values are set to TSG3nCMP1E to TSG3nCMP4E registers according to the formulas described in **Section 25.4.7.6, Compare Register Setting in HSP-PWM Mode.**

When this register is read, the same value as TSG3nCMP1E is returned.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 134_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nHSPCMUE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nHSPCMUE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.58 TSG3nHSPCMVE — TSG3n HSP-PWM Mode V Phase Compare Register

This register sets the PWM output width for V phase in HSP-PWM mode.

When a write access is made to this register, the values are set to TSG3nCMP5E to TSG3nCMP8E registers according to the formulas described in **Section 25.4.7.6, Compare Register Setting in HSP-PWM Mode.**

When this register is read, the same value as TSG3nCMP5E is returned.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 130_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nHSPCMVE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nHSPCMVE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.59 TSG3nHSPCMWE — TSG3n HSP-PWM Mode W Phase Compare Register

This register sets the PWM output width for W phase in HSP-PWM mode.

When a write access is made to this register, the values are set to TSG3nCMP9E to TSG3nCMP12E registers according to the formulas described in **Section 25.4.7.6, Compare Register Setting in HSP-PWM Mode.**

When this register is read, the same value as TSG3nCMP9E is returned.

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 12C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nHSPCMWE (18-bit compare register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nHSPCMWE (18-bit compare register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.60 TSG3nHSPSHUE — TSG3n HSP-PWM Mode U Phase Shift Register

This register sets the PWM shift width for U phase in HSP-PWM mode.

When a write access is made to TSG3nHSPCMUE after setting the TSG3nHSPSHUE register, the values are set to TSG3nCMP1E to TSG3nCMP4E registers according to the formulas described in **Section 25.4.7.6, Compare Register Setting in HSP-PWM Mode.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 128_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------------------------------|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nHSPSHUE (18-bit shift register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nHSPSHUE (18-bit shift register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.61 TSG3nHSPSHVE — TSG3n HSP-PWM Mode V Phase Shift Mode Register

This register sets the PWM shift width for V phase in HSP-PWM mode.

When a write access is made to TSG3nHSPCMVE after setting the TSG3nHSPSHVE register, the values are set to TSG3nCMP5E to TSG3nCMP8E registers according to the formulas described in **Section 25.4.7.6, Compare Register Setting in HSP-PWM Mode.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 124_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------------------------------|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nHSPSHVE (18-bit shift register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nHSPSHVE (18-bit shift register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.62 TSG3nHSPSHWE — TSG3n HSP-PWM Mode W Phase Shift Register

This register sets the PWM shift width for U phase in HSP-PWM mode.

When a write access is made to TSG3nHSPCMWE after setting the TSG3nHSPSHWE register, the values are set to TSG3nCMP9E to TSG3nCMP12E registers according to the formulas described in **Section 25.4.7.6, Compare Register Setting in HSP-PWM Mode.**

Access: This register can be read/written in 32-bit units.

Address: <TSG3n_base> + 120_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------------------------------|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSG3nHSPSHWE (18-bit shift register) | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TSG3nHSPSHWE (18-bit shift register) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

25.3.63 TSG3nDTPR — TSG3n Dead Time Protection Register

This register controls protection of the write access to the dead time register.

Access: This register can be read/written in 16-bit units.

Address: <TSG3n_base> + 210_H

Value after reset: 0000 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | TSG3nDTPR (Write Protection Code) | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 25.36 TSG3nDTPR Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 15 | TSG3nDTCM | Enables or disables rewriting of TSG3nDTC0 and TSG3nDTC1. 0: Enables rewriting of TSG3nDTC0 and TSG3nDTC1. 1: Disables rewriting of TSG3nDTC0 and TSG3nDTC1. |
| 14 to 0 | TSG3nDTPR [14:0] | Sets the write protection code (any value from 0000 to 7FFF). |

This register protects TSG3nDTC0 and TSG3nDTC1 from illegal rewriting.

Functions are described below.

- TSG3nDTCM enables or disables rewriting of TSG3nDTC0 and TSG3nDTC1.
- Rewriting of TSG3nDTC0 and TSG3nDTC1 is enabled or disabled by double-checking a match of the write protection code (bits 30 to 16) of TSG3nDTC0 and TSG3nDTC1 with the write protection code of TSG3nDTPR, and the TSG3nDTCM setting.

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

25.4 Function

Table 25.37 List of Modes

| TSG3nCTL0 Register | | | Timer Mode |
|--------------------|----------|----------|--|
| TSG3nMD2 | TSG3nMD1 | TSG3nMD0 | |
| 0 | 0 | 0 | PWM mode |
| 0 | 0 | 1 | HT-PWM mode (HT-PWM) |
| 0 | 1 | 0 | Shift pulse PWM mode (SP-PWM) |
| 0 | 1 | 1 | 120-DC mode |
| 1 | 0 | 0 | High-accuracy shift pulse PWM mode (HSP-PWM) |
| Other than above | | | Setting prohibited |

25.4.1 Basic Operation

25.4.1.1 Basic Operation of 18-Bit Counter

The basic operation of the 18-bit counter is described. For details, see **Section 25.4.7, Operating Modes**.

Counting start

The 18-bit counter of TSG3n starts counting in HT-PWM mode when the initial value is 00000_H and after the TSG3nDTC0 value is loaded. The counter starts counting from the initial value 00000_H in all modes except for HT-PWM mode.

In HT-PWM mode, the counter value is incremented by 2 from the value of TSG3nDTC0 and decremented by 2 down to the value of TSG3nDTC0 after the counter value matches with the value of TSG3nCMP0E + TSG3nDTC0. The counter increments from 00000_H, 00001_H, 00002_H, 00003_H, ... in all modes except for HT-PWM mode.

Counter clear

The 18-bit counter is cleared by the match of the counter value and the value of TSG3nCMP0E in all modes except for HT-PWM mode. (Clearing operation is not available in HT-PWM mode.)

Counter read during counting

In the TSG3n, the 18-bit counter value during counting can be read through TSG3nCNTL.

Count stop operation

When count operation is stopped (when TS0TE is changed from 1 to 0), TSG3nCNTL and TSG3nSBCE retain the counter value when stopped.

Interrupt operation

In the TSG3n, the following interrupts are generated.

- INTTSG3nI0: A period interrupt by a match of the 18-bit counter value with the TSG3nDTC0 value in HT-PWM mode. A compare match interrupt of the 18-bit counter value with the TSG3nCMP0E buffer register in any mode other than HT-PWM mode.
- INTTSG3nI1: A compare match interrupt of the 18-bit counter value with the TSG3nCMP1E buffer register.
- INTTSG3nI2: A compare match interrupt of the 18-bit counter value with the TSG3nCMP2E buffer register.
- INTTSG3nI3: A compare match interrupt of the 18-bit counter value with the TSG3nCMP3E buffer register.
- INTTSG3nI4: A compare match interrupt of the 18-bit counter value with the TSG3nCMP4E buffer register.
- INTTSG3nI5: A compare match interrupt of the 18-bit counter value with the TSG3nCMP5E buffer register.
- INTTSG3nI6: A compare match interrupt of the 18-bit counter value with the TSG3nCMP6E buffer register.
- INTTSG3nI7: A compare match interrupt of the 18-bit counter value with the TSG3nCMP7E buffer register.
- INTTSG3nI8: A compare match interrupt of the 18-bit counter value with the TSG3nCMP8E buffer register.
- INTTSG3nI9: A compare match interrupt of the 18-bit counter value with the TSG3nCMP9E buffer register.
- INTTSG3nI10: A compare match interrupt of the 18-bit counter value with the TSG3nCMP10E buffer register.
- INTTSG3nI11: A compare match interrupt of the 18-bit counter value with the TSG3nCMP11E buffer register.
- INTTSG3nI12: A compare match interrupt of the 18-bit counter value with the TSG3nCMP12E buffer register.
- INTTSG3nIPEK: A peak interrupt when the 18-bit counter switches from incrementing to decrementing.
- INTTSG3nIVLY: A valley interrupt when the 18-bit counter switches from decrementing to incrementing.
- INTTSG3nIER: A simultaneous active state detection interrupt of the positive phase and inverse phase
- INTTSG3nIWN: A warning detection interrupt

25.4.1.2 Function of Compare Registers

The functions of the compare registers in each operating mode are shown in the following tables.

Table 25.38 Compare Register Functions in Each Mode (1/7)

| Operating Mode | TSG3nCMP0E | TSG3nCMP1E | TSG3nCMP2E |
|----------------|------------|--|--|
| PWM mode | PWM period | TSG3nO1 clear timing | TSG3nO1 set timing |
| HT-PWM mode | PWM period | TSG3nO1 clear timing TSG3nO2 set timing | TSG3nO1 set timing TSG3nO2 clear timing |
| SP-PWM mode | PWM period | TSG3nO1 clear timing TSG3nO2 set timing | TSG3nO1 set timing TSG3nO2 clear timing |
| 120-DC mode | PWM period | Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0 | Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0 |
| HSP-PWM mode | PWM period | TSG3nO1 clear timing | TSG3nO1 set timing |

Table 25.38 Compare Register Functions in Each Mode (2/7)

| Operating Mode | TSG3nCMP3E | TSG3nCMP4E | TSG3nCMP5E | TSG3nCMP6E |
|----------------|--|--|--|--|
| PWM mode | TSG3nO2 clear timing | TSG3nO2 set timing | TSG3nO3 clear timing | TSG3nO3 set timing |
| HT-PWM mode | Compare match interrupt | Compare match interrupt | TSG3nO3 clear timing TSG3nO4 set timing | TSG3nO3 set timing TSG3nO4 clear timing |
| SP-PWM mode | — | — | TSG3nO3 clear timing TSG3nO4 set timing | TSG3nO3 set timing TSG3nO4 clear timing |
| 120-DC mode | Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W | Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W | Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W | Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W |
| HSP-PWM mode | TSG3nO2 clear timing | TSG3nO2 set timing | TSG3nO3 clear timing | TSG3nO3 set timing |

Table 25.38 Compare Register Functions in Each Mode (3/7)

| Operating Mode | TSG3nCMP7E | TSG3nCMP8E | TSG3nCMP9E | TSG3nCMP10E |
|----------------|--|--|--|---|
| PWM mode | TSG3nO4 clear timing | TSG3nO4 set timing | TSG3nO5 clear timing | TSG3nO5 set timing |
| HT-PWM mode | Compare match interrupt | Compare match interrupt | TSG3nO5 clear timing TSG3nO6 set timing | TSG3nO5 set timing TSG3nO6 clear timing |
| SP-PWM mode | — | — | TSG3nO5 clear timing TSG3nO6 set timing | TSG3nO5 set timing TSG3nO6 clear timing |
| 120-DC mode | Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W | Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W | Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W | TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W |
| HSP-PWM mode | TSG3nO4 clear timing | TSG3nO4 set timing | TSG3nO5 clear timing | TSG3nO5 set timing |

Table 25.38 Compare Register Functions in Each Mode (4/7)

| Operating Mode | TSG3nCMP11E | TSG3nCMP12E | TSG3nDCMP0E | TSG3nDCMP1E |
|----------------|---|---|---|---|
| PWM mode | TSG3nO6 clear timing | TSG3nO6 set timing | Diagnostic output or A/D conversion trigger timing | Diagnostic output or A/D conversion trigger timing |
| HT-PWM mode | Compare match interrupt | Compare match interrupt | Diagnostic output or A/D conversion trigger timing | Diagnostic output or A/D conversion trigger timing |
| SP-PWM mode | — | — | Diagnostic output or A/D conversion trigger timing | Diagnostic output or A/D conversion trigger timing |
| 120-DC mode | Select TSG3nO2, TSG3nO4, TSG3nO6 by TSG3nPAT1W | Select TSG3nO2, TSG3nO4, TSG3nO6 by TSG3nPAT1W | Diagnostic output or A/D conversion trigger timing | Diagnostic output or A/D conversion trigger timing |
| HSP-PWM mode | TSG3nO6 clear timing | TSG3nO6 set timing | Diagnostic output or A/D conversion trigger timing | Diagnostic output or A/D conversion trigger timing |

Table 25.38 Compare Register Functions in Each Mode (5/7)

| Operating Mode | TSG3nDCMP2E | TSG3nCMPUE | TSG3nCMPVE | TSG3nCMPWE |
|----------------|--|---|---|--|
| PWM mode | Diagnostic output or A/D conversion trigger timing | — | — | — |
| HT-PWM mode | Diagnostic output or A/D conversion trigger timing | The TSG3nCMPUE set value is used as the set value of TSG3nCMP1E and TSG3nCMP2E. | The TSG3nCMPVE set value is used as the set value of TSG3nCMP5E and TSG3nCMP6E. | The TSG3nCMPWE set value is used as the set value of TSG3nCMP9E and TSG3nCMP10E. |
| SP-PWM mode | Diagnostic output or A/D conversion trigger timing | — | — | — |
| 120-DC mode | Diagnostic output or A/D conversion trigger timing | — | — | — |
| HSP-PWM mode | Diagnostic output or A/D conversion trigger timing | — | — | — |

Table 25.38 Compare Register Functions in Each Mode (6/7)

| Operating Mode | TSG3nUPWE | TSG3nVPWE | TSG3nWPWE |
|----------------|--|--|--|
| PWM mode | — | — | — |
| HT-PWM mode | — | — | — |
| SP-PWM mode | The sum of the TSG3nUPWE set value and the TSG3nCMP2E set value is used as the TSG3nCMP1E set value. | The sum of the TSG3nVPWE set value and the TSG3nCMP2E set value is used as the TSG3nCMP1E set value. | The sum of the TSG3nWPWE set value and the TSG3nCMP2E set value is used as the TSG3nCMP1E set value. |
| 120-DC mode | — | — | — |
| HSP-PWM mode | — | — | — |

Table 25.38 Compare Register Functions in Each Mode (7/7)

| Operating Mode | TSG3nHSPCMUE, TSG3nHSPSHUE | TSG3nHSPCMVE, TSG3nHSPSHVE | TSG3nHSPCMWE, TSG3nHSPSHWE |
|----------------|--|--|---|
| PWM mode | — | — | — |
| HT-PWM mode | — | — | — |
| SP-PWM mode | — | — | — |
| 120-DC mode | — | — | — |
| HSP-PWM mode | TSG3nCMP1E-4E is set based on the TSG3nHSPCMUE set value and the set values in TSG3nCMP0E, TSG3nDTC0, TSG3nDTC1, and TSG3nHSPSHUE. | TSG3nCMP5E-8E is set based on the TSG3nHSPCMVE set value and the set values of TSG3nCMP0E, TSG3nDTC0, TSG3nDTC1, and TSG3nHSPSHVE. | TSG3nCMP9E-12E is set based on the TSG3nHSPCMWE set value and the set values of TSG3nCMP0E, TSG3nDTC0, TSG3nDTC1, and TSG3nHSPSHWE. |

25.4.1.3 Compare Register Rewrite Operation

TSG3 can be set to reload mode or anytime rewrite mode using the TSG3nRMC bit.

Reload mode is enabled when TSG3nRMC = 0. The registers listed as “Enable” on the “Reload” column in **Section 25.3.1, List of Registers** are simultaneously updated at the reload timing.

Anytime rewrite mode is enabled when TSG3nRMC = 1. The registers are updated independently every time the value is written to the relevant register.

The update timing of the registers to be reloaded in reload mode and anytime rewrite mode in each mode are listed in the following table.

Table 25.39 Updating Timing of Compare Registers by Mode

| Mode | Anytime Rewrite TSG3nRMC = 1 | Reload TSG3nRMC = 0 |
|--------------|--|------------------------|
| PWM mode | TSG3nCMP0E: At the next counter clear timing of the 18-bit counter | At reload timing |
| | Registers other than TSG3nCMP0E: At a write access to the register | |
| HT-PWM mode | TSG3nCMP0E: At the next peak or valley timing of TSG3nCnTE | At reload timing |
| | TSG3nCMP1E, 2E, 5E, 6E, 9E, 10E: At writing 1 to the TSG3nIMT bit | |
| | Registers other than TSG3nCMP0E,1E, 2E, 5E, 6E, 9E, 10E: At a write access to the register | |
| SP-PWM mode | TSG3nCMP0E: At the next counter clear timing of the 18-bit counter | At reload timing |
| | Registers other than TSG3nCMP0E: At a write access to the register | |
| 120-DC mode | Setting prohibited | At reload timing |
| HSP-PWM mode | Setting prohibited | At reload timing |

Anytime Rewrite Mode

In this mode, the compare registers are rewritten independently. Whenever a value is written to the compare register, the written value is reflected at the timing of **Table 25.39**.

Reload mode (Simultaneous Rewrite Function)

Writing to TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE) enables reload (sets the reload request flag (TSG3nSTR0.TSG3nRSF)), and the values of all the pertinent registers are updated simultaneously at the next reload timing (reload).

The reload timing is the peak or valley timing of the 18-bit counter when the TSG3nTRG0.TSG3nTS bit is changed from 0 to 1. Reloading is controlled by TSG3nCTL4.TSG3nPRE and TSG3nVRE.

Writing to any register other than TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE) does not enable reloading.

Do not write to the registers to be reloaded until the next reload timing after reloading is enabled by writing to TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE). The pertinent registers should be rewritten when the reload request flag (TSG3nSTR0.TSG3nRSF) is 0.

Rewriting registers to be reloaded by DMA transfer

Some of the registers to be reloaded can be rewritten by DMA transfer. DMA transfer is performed as follows.

Table 25.40 Example of DMA Transfer Order of Registers to be Reloaded


| Address | Register Name | DMA Transfer Order (Example) |
|---------------------------------|---------------|---|
| <TSG3n_base> + 040 _H | TSG3nCMP1W |  |
| <TSG3n_base> + 044 _H | TSG3nCMP5W | |
| <TSG3n_base> + 048 _H | TSG3nCMP9W | |
| <TSG3n_base> + 04C _H | TSG3nCMP3W | |
| <TSG3n_base> + 050 _H | TSG3nCMP7W | |
| <TSG3n_base> + 054 _H | TSG3nCMP11W | |
| <TSG3n_base> + 058 _H | TSG3nCMP0 | |
| <TSG3n_base> + 05C _H | TSG3nDCMP0W | |
| <TSG3n_base> + 060 _H | TSG3nDCMP2 | |
| <TSG3n_base> + 064 _H | TSG3nPAT0W | |
| <TSG3n_base> + 068 _H | TSG3nPAT1W | |
| <TSG3n_base> + 06C _H | TSG3nDTC0W | |
| <TSG3n_base> + 070 _H | TSG3nDTC1W | |

Table 25.41 Example of DMA Transfer Order of Registers to be Reloaded


| Address | Register Name | DMA Transfer Order (Example) |
|---------------------------------|---------------|---|
| <TSG3n_base> + 140 _H | TSG3nDCMP2E |  |
| <TSG3n_base> + 144 _H | TSG3nDCMP1E | |
| <TSG3n_base> + 148 _H | TSG3nDCMP0E | |
| <TSG3n_base> + 14C _H | TSG3nCMP0E | |
| <TSG3n_base> + 150 _H | TSG3nCMP12E | |
| <TSG3n_base> + 154 _H | TSG3nCMP11E | |
| <TSG3n_base> + 158 _H | TSG3nCMP8E | |
| <TSG3n_base> + 15C _H | TSG3nCMP7E | |
| <TSG3n_base> + 160 _H | TSG3nCMP4E | |
| <TSG3n_base> + 164 _H | TSG3nCMP3E | |
| <TSG3n_base> + 168 _H | TSG3nCMP10E | |
| <TSG3n_base> + 16C _H | TSG3nCMP9E | |
| <TSG3n_base> + 170 _H | TSG3nCMP6E | |
| <TSG3n_base> + 174 _H | TSG3nCMP5E | |
| <TSG3n_base> + 178 _H | TSG3nCMP2E | |
| <TSG3n_base> + 17C _H | TSG3nCMP1E | |

Table 25.42 Duty Setting in HT-PWM Mode


| Address | Register Name | DMA Transfer Order (Example) |
|---------------------------------|---------------|---|
| <TSG3n_base> + 180 _H | TSG3nCMPWE |  |
| <TSG3n_base> + 184 _H | TSG3nCMPVE | |
| <TSG3n_base> + 188 _H | TSG3nCMPUE | |

Table 25.43 Active Width Setting in SP-PWM Mode

| Address | Register Name | DMA Transfer Order (Example) |
|---------------------------------|---------------|------------------------------|
| <TSG3n_base> + 190 _H | TSG3nWPWE | ↓ |
| <TSG3n_base> + 194 _H | TSG3nVPWE | |
| <TSG3n_base> + 198 _H | TSG3nUPWE | |

Table 25.44 Shift Width and Duty Setting in HSP-PWM Mode

| Address | Register Name | DMA Transfer Order (Example) |
|---------------------------------|---------------|------------------------------|
| <TSG3n_base> + 120 _H | TSG3nHSPSHWE | ↓ |
| <TSG3n_base> + 124 _H | TSG3nHSPSHVE | |
| <TSG3n_base> + 128 _H | TSG3nHSPSHUE | |
| <TSG3n_base> + 12C _H | TSG3nHSPCMWE | |
| <TSG3n_base> + 130 _H | TSG3nHSPCMVE | |
| <TSG3n_base> + 134 _H | TSG3nHSPCMUE | |

NOTES

1. TSG3nCTL4 and TSG3nIOC3 should be rewritten individually.
2. Since writing to TSG3nCMP1E (including TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE) enables reloading, it should be rewritten after all the other registers to be reloaded have been rewritten (ready to be reloaded)

(1) Example of Operation in Anytime Rewrite Mode

In this mode, the values written to the compare registers (TSG3nCMP1E to TSG3nCMP12E) are transferred to the internal buffer registers immediately, and are compared with the counter value.

The values are transferred to the internal compare buffer registers one clock cycle (PCLK) after being written to the compare registers (TSG3nCMP1E to TSG3nCMP12E).

The transfer timing of the TSG3nCMP0E is the peak or valley timing (only in HT-PWM mode) of the 18-bit counter after being written to the compare registers, or at the match timing of the TSG3nCMP0E value with the 18-bit counter value (in any mode other than HT-PWM mode).

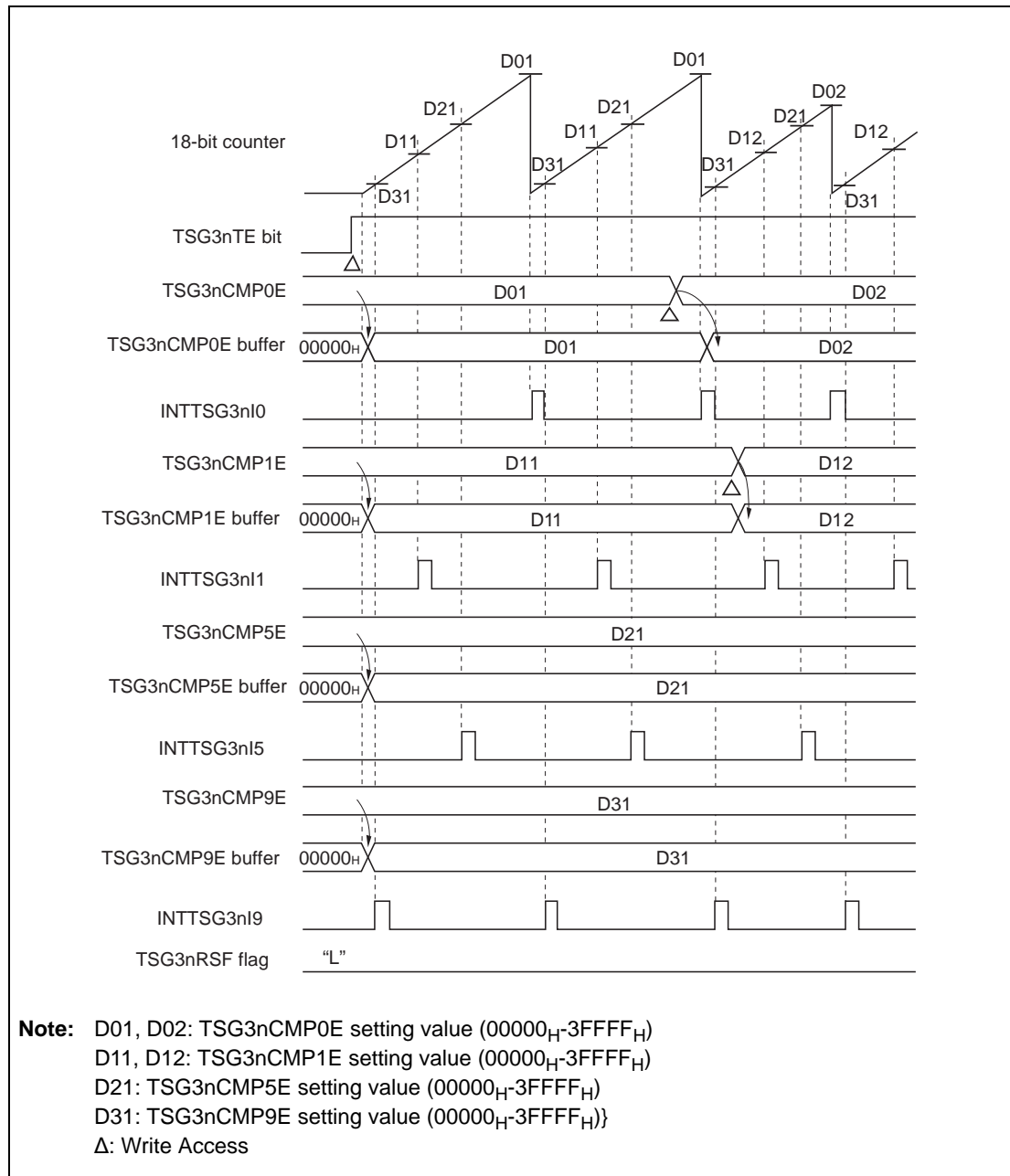


Figure 25.8 Anytime Rewrite Timing (Example in PWM Mode)

(a) Data reflection on PWM in Anytime Rewrite in HT-PWM

In anytime rewrite operation in HT-PWM mode, the values are transferred to the buffer at the timing when 1 is written to the TSG3nIMT bit after the settings of TSG3nCMP1E, 2E, 5E, 6E, 9E and 10E registers are modified and PWM output is forcibly set/cleared depending on the modified set value.

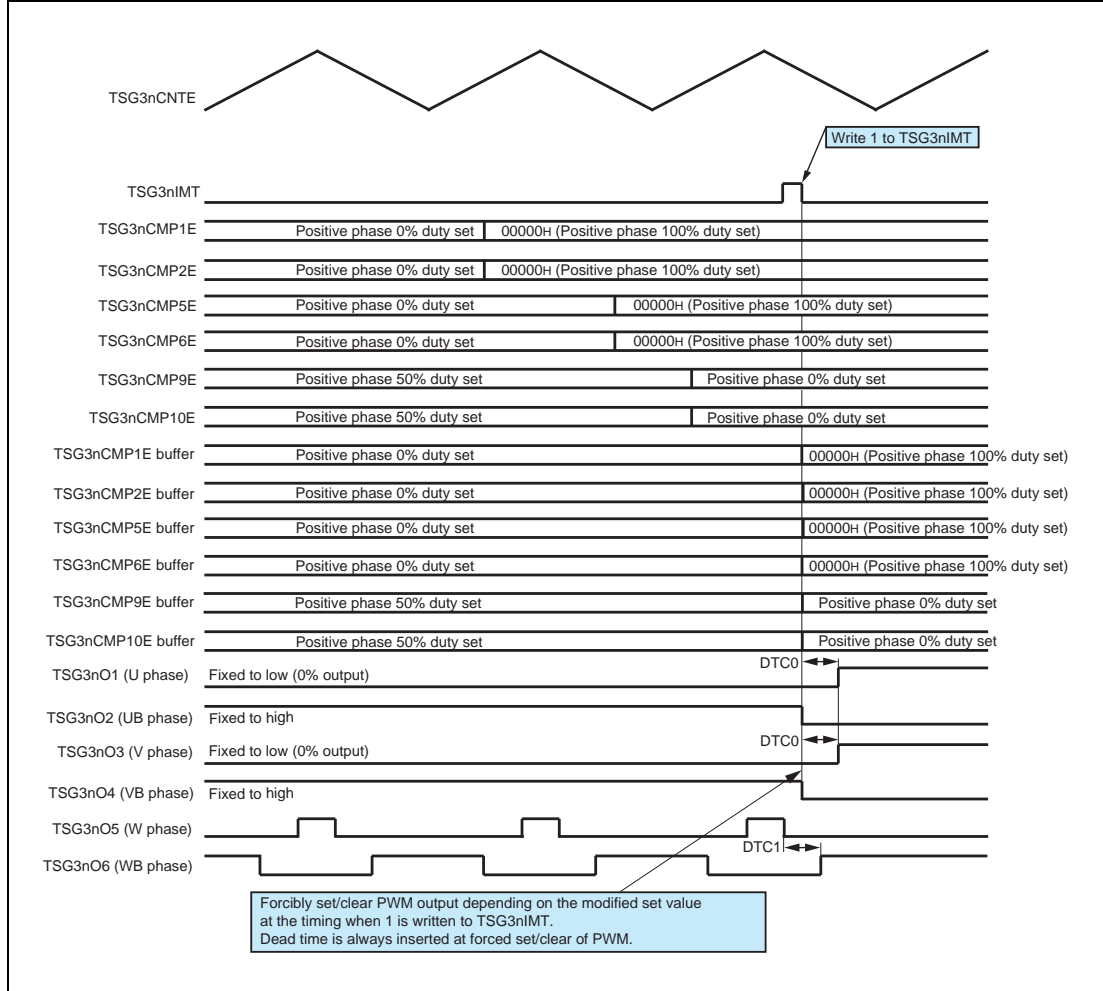


Figure 25.9 Update Timing of TSG3nCMP1E, 2E, 5E, 6E, 9E, and 10E at Anytime Rewrite Operation in HT-PWM Mode

(2) Example of Operation in Reload Mode (Simultaneous Rewrite Function)

The rewritten values of the registers to be reloaded (the registers listed in the **Section 25.3.1, List of Registers** with “Enabled” in the column of “Reload”) can be transferred to the corresponding buffer registers simultaneously at the reload timing.

The registers should be rewritten when the pertinent reload request flag (TSG3nSTR0.TSG3nRSF) is 0.

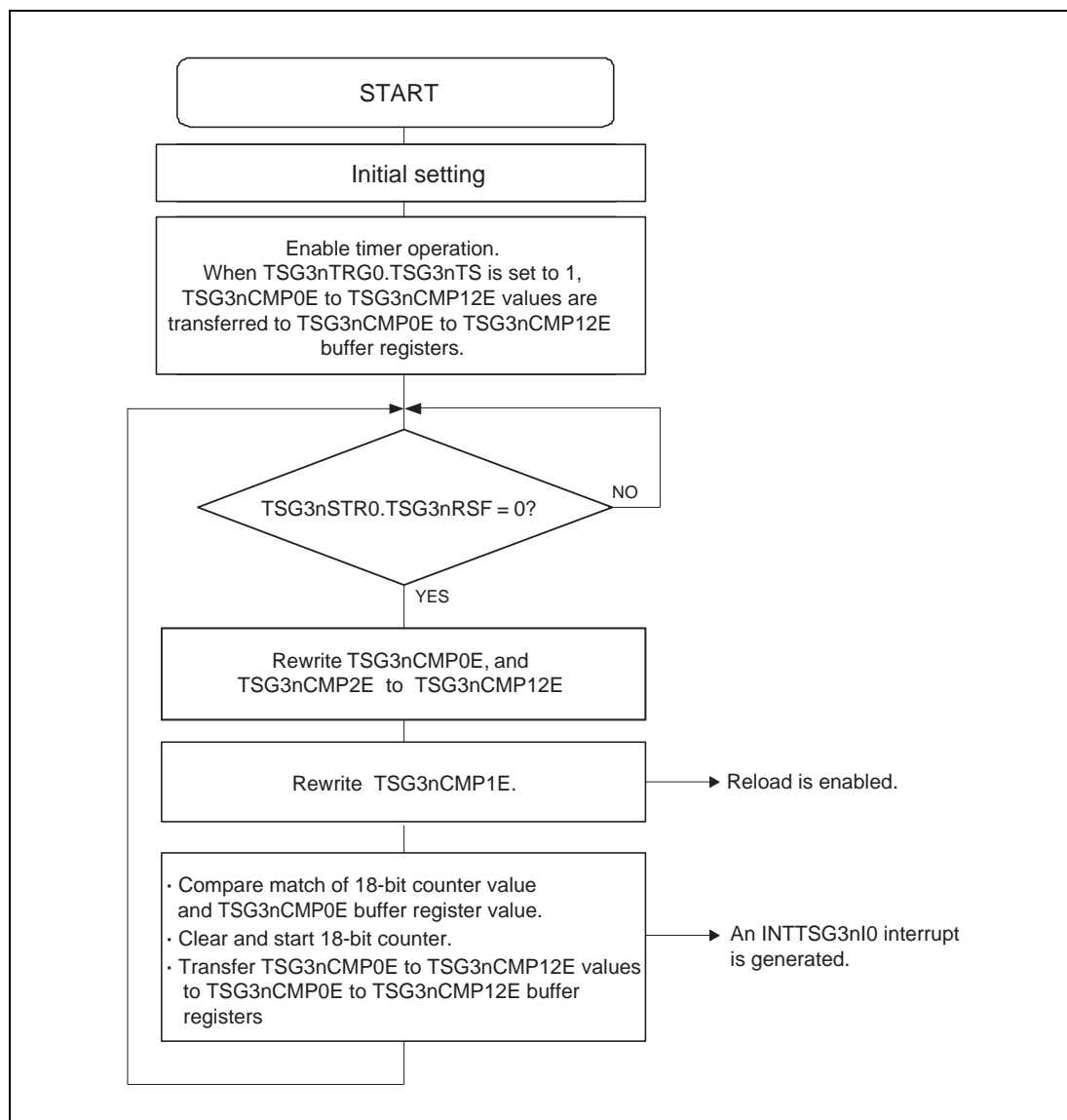


Figure 25.10 Basic Operation Flow in Reload Mode (Simultaneous Rewrite Function) (Example of PWM Mode)

CAUTION

Writing to TSG3nCMP1E also enables reloading. Therefore, TSG3nCMP1E should be rewritten after TSG3nCMP0E and TSG3nCMP2E to TSG3nCMP12E registers have been rewritten.

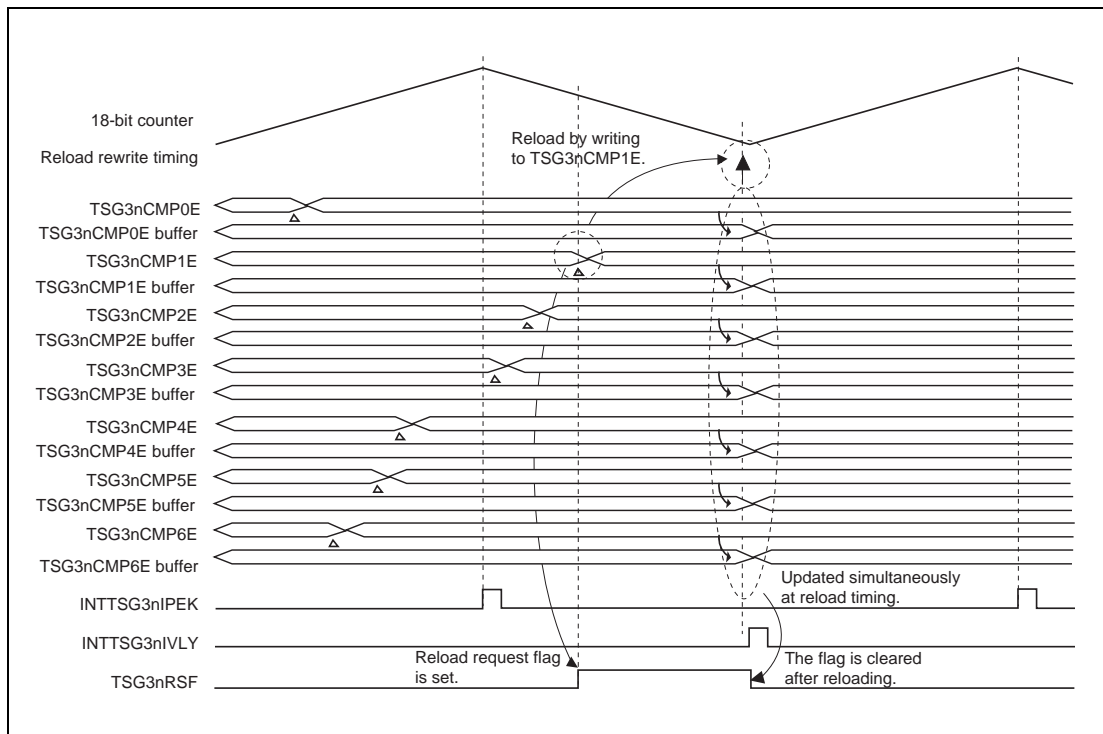


Figure 25.11 Simultaneous Rewrite Timing (Example of HT-PWM Mode) (1/2)

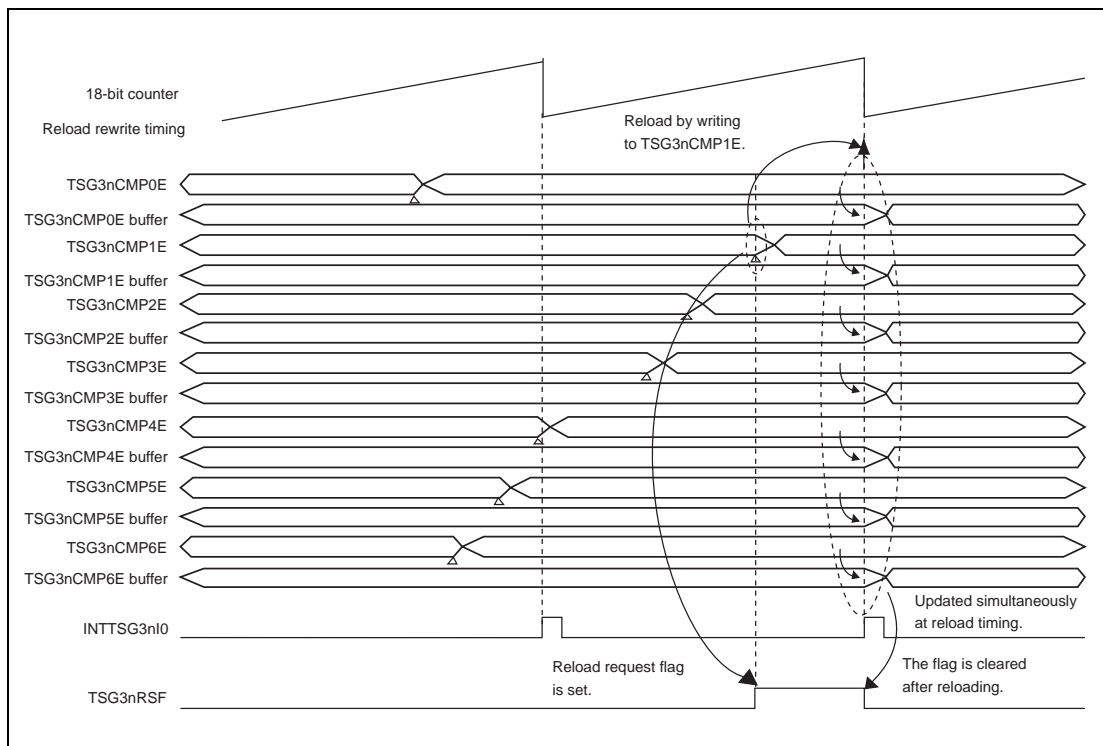


Figure 25.11 Simultaneous Rewrite Timing (Example of PWM Mode) (2/2)

(a) Reload Rewrite Setting Example in Each Mode

Reloading conditions and setting examples are shown in the following tables.

Table 25.45 List of Reload Settings (when TSG3nCTL3.TSG3nRIA = 0)

| Mode | TSG3nCTL4. TSG3nPRE | TSG3nCTL4. TSG3nVRE | TSG3nCTL4. TSG3nPIE | TSG3nCTL4. TSG3nVIE | TSG3nCTL4. TSG3nRCC04- TSG3nRCC00 | Reload |
|--------------|------------------------|------------------------|------------------------|------------------------|---|--|
| PWM mode | 0 | 0/1 | 0/1 | 0/1 | Any value | Setting prohibited |
| SP-PWM mode | 1 | 0 | 0/1 | 0/1 | Any value | When INTTSG3nI0 is generated. |
| 120-DC mode | 1 | 1 | 0/1 | 0/1 | Any value | When INTTSG3nI0 is generated. |
| HSP-PWM mode | 1 | 1 | 0/1 | 0/1 | Any value | When INTTSG3nI0 is generated. |
| HT-PWM mode | 0 | 0 | 0/1 | 0/1 | Any value | Setting prohibited |
| | 0 | 1 | 0/1 | 0/1 | Any value | When INTTSG3nIVLY is generated. |
| | 1 | 0 | 0/1 | 0/1 | Any value | When INTTSG3nIPEK is generated |
| | 1 | 1 | 0/1 | 0/1 | Any value | When INTTSG3nIPEK or INTTSG3nIVLY is generated. |

Table 25.46 List of Reload Settings (when TSG3nCTL3.TSG3nRIA = 1)

| Mode | TSG3nCTL4. TSG3nPRE | TSG3nCTL4. TSG3nVRE | TSG3nCTL4. TSG3nPIE | TSG3nCTL4. TSG3nVIE | TSG3nCTL4. TSG3nRCC04- TSG3nRCC00 | Reload |
|--------------|------------------------|------------------------|------------------------|------------------------|---|---|
| PWM mode | 0 | 0/1 | 0/1 | 0/1 | Any value | Setting prohibited |
| SP-PWM mode | 1 | 0 | 0 | 0/1 | Any value | Setting prohibited |
| 120-DC mode | 1 | 0 | 1 | 0/1 | Any value | When INTTSG3nI0 is generated |
| HSP-PWM mode | 1 | 1 | 0 | 0/1 | Any value | Setting prohibited |
| | 1 | 1 | 1 | 0/1 | Any value | When INTTSG3nI0 is generated |
| HT-PWM mode | 0 | 0 | 0/1 | 0/1 | Any value | Setting prohibited |
| | 0 | 1 | 0 | 0 | Any value | Setting prohibited |
| | 0 | 1 | 0 | 1 | Any value | When INTTSG3nIVLY is generated |
| | 0 | 1 | 1 | 0 | Any value | Setting prohibited |
| | 0 | 1 | 1 | 1 | Any value | When INTTSG3nIVLY is generated |
| | 1 | 0 | 0 | 0/1 | Any value | Setting prohibited |
| | 1 | 0 | 1 | 0/1 | Any value | When INTTSG3nIPEK is generated |
| | 1 | 1 | 0 | 0 | Any value | Setting prohibited |
| | 1 | 1 | 0 | 1 | Any value | When INTTSG3nIVLY is generated |
| | 1 | 1 | 1 | 0 | Any value | When INTTSG3nIPEK is generated |
| | 1 | 1 | 1 | 1 | Any value | When INTTSG3nIPEK or INTTSG3nIVLY is generated |

25.4.1.4 List of Outputs in Each Mode

The list of timer outputs (TSG3nO0-7 pins) in each mode is shown in the following tables.

Table 25.47 List of Timer Outputs in Each mode (1/3)

| Operating Mode | TSG3nO0 Pin | TSG3nO1 Pin | TSG3nO2 Pin |
|----------------|---|--|---|
| PWM mode | — (Fixed to low.) | Outputs a PWM signal by compare match of TSG3nCMP1E and TSG3nCMP2E. | Outputs a PWM signal by compare match of TSG3nCMP3E and TSG3nCMP4E. |
| HT-PWM mode | Outputs the status indicating whether the 18-bit counter or 18-bit sub-counter is incremented or decremented. | Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP1E and TSG3nCMP2E. | Outputs an inverse phase PWM signal (with dead time) to TSG3nO1 pin. |
| SP-PWM mode | — (Fixed to low.) | Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP1E and TSG3nCMP2E. | Outputs an inverse phase PWM signal (with dead time) to TSG3nO1. |
| 120-DC mode | — (Fixed to low.) | Outputs a PWM signal by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E. | Outputs a PWM signal by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E. |
| HSP-PWM mode | — (Fixed to low.) | Outputs a PWM signal by compare match of TSG3nCMP1E and TSG3nCMP2E. | Outputs a PWM signal by compare match of TSG3nCMP3E and TSG3nCMP4E. |

Table 25.47 List of Timer Outputs in Each mode (2/3)

| Operating Mode | TSG3nO3 Pin | TSG3nO4 Pin | TSG3nO5 Pin |
|----------------|--|---|--|
| PWM mode | Outputs a PWM signal by compare match of TSG3nCMP5E and TSG3nCMP6E. | Outputs a PWM signal by compare match of TSG3nCMP7E and TSG3nCMP8E. | Outputs a PWM signal by compare match of TSG3nCMP9E and TSG3nCMP10E. |
| HT-PWM mode | Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP5E and TSG3nCMP6E. | Outputs an inverse phase PWM signal (with dead time) to TSG3nO3. | Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP9E and TSG3nCMP10E. |
| SP-PWM mode | Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP5E and TSG3nCMP6E. | Outputs an inverse phase PWM signal (with dead time) to TSG3nO3. | Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP9E and TSG3nCMP10E. |
| 120-DC mode | Outputs a PWM signal by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E. | Outputs a PWM signal by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E. | Outputs a PWM signal by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E. |
| HSP-PWM mode | Outputs a PWM signal by compare match of TSG3nCMP5E and TSG3nCMP6E. | Outputs a PWM signal by compare match of TSG3nCMP7E and TSG3nCMP8E. | Outputs a PWM signal by compare match of TSG3nCMP9E and TSG3nCMP10E. |

Table 25.47 List of Timer Outputs in Each mode (3/3)

| Operating Mode | TSG3nO6 Pin | TSG3nO7 Pin |
|----------------|---|--|
| PWM mode | Outputs a PWM signal by compare match of TSG3nCMP11E and TSG3nCMP12E. | Outputs a diagnostic signal or A/D conversion trigger.* ¹ |
| HT-PWM mode | Outputs an inverse phase PWM signal (with dead time) to TSG3nO5. | Outputs a diagnostic signal or A/D conversion trigger.* ¹ |
| SP-PWM mode | Outputs an inverse phase PWM signal (with dead time) to TSG3nO5. | Outputs a diagnostic signal or A/D conversion trigger.* ¹ |
| 120-DC mode | Outputs a PWM signal by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E. | Outputs a diagnostic signal or A/D conversion trigger.* ¹ |
| HSP-PWM mode | Outputs a PWM signal by compare match of TSG3nCMP11E and TSG3nCMP12E. | Outputs a diagnostic signal or A/D conversion trigger.* ¹ |

Note 1. For TSG3nO7, See **Section 25.4.1.4 (a), TSG3nO7 Pin Output Control**

(a) TSG3nO7 Pin Output Control

The TSG3nO7 pin can output a pulse of A/D conversion trigger (TSG3nIOC1.TSG3nTGS = 0) or diagnostic output (TSG3nIOC1.TSG3nTGS = 1). When outputting a pulse of A/D conversion trigger, the TSG3nO7 pin is activated at the rising edge of the TSG3nADTRG0 signal, and inactivated at the rising edge of the TSG3nADTRG1 signal. When the TSG3nADTRG0 signal is detected while the TSG3nO7 pin is active, the TSG3nO7 pin remains active. When the TSG3nADTRG1 signal is detected while the TSG3nO7 pin is inactive, the TSG3nO7 pin remains inactive. If TSG3nADTRG0 and TSG3nADTRG1 signal triggers occur simultaneously, the TSG3nO7 pin is inactivated.

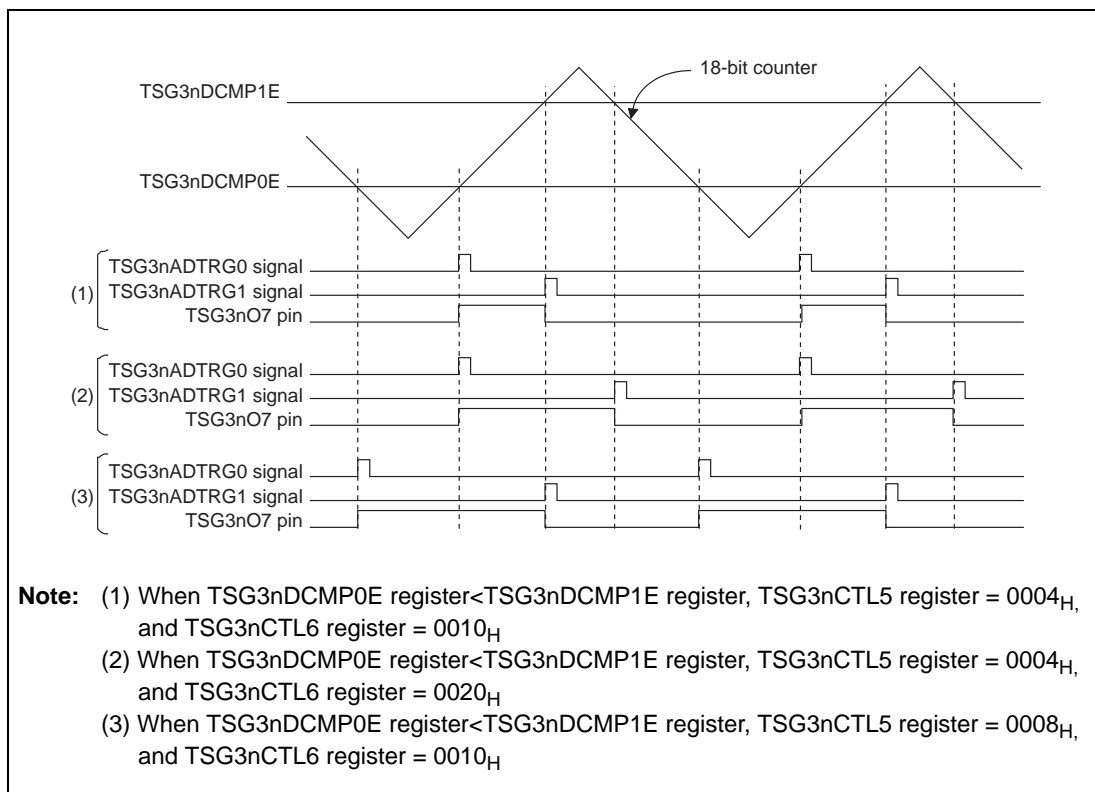


Figure 25.12 Example of A/D Trigger Output Timing of TSG3nO7 Pin (TSG3nIOC1.TSG3nTGS = 0)

During diagnostic output, the active level is output on the TSG3nO7 pin with the width specified by the TSG3nCTL0.TSG3nDWD bit when the values of the TSG3nDCMP0E to TSG3nDCMP2E bits match that of the 18-bit counter. If a TSG3nDCMP0E to TSG3nDCMP2E value again matches the value of the 18-bit counter value while the diagnostic output on the TSG3nO7 pin is already at the active level, pulse output on the TSG3nO7 pin continues for the number of cycles of PCLK (8 or 16) set by the TSG3nDWD bit from the point of the later match.

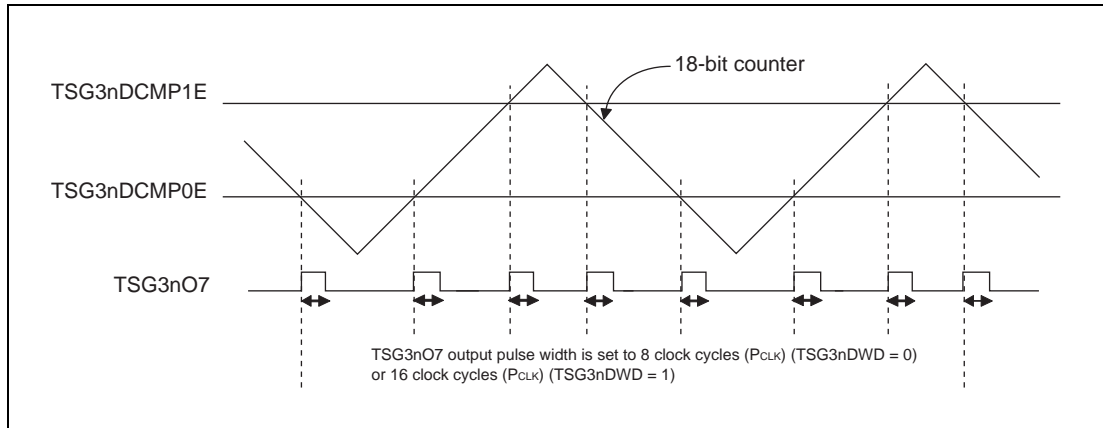


Figure 25.13 Example of TSG3nO7 Pin Diagnostic Pulse Output Timing (1) (TSG3nIOC1.TSG3nTGS = 1)

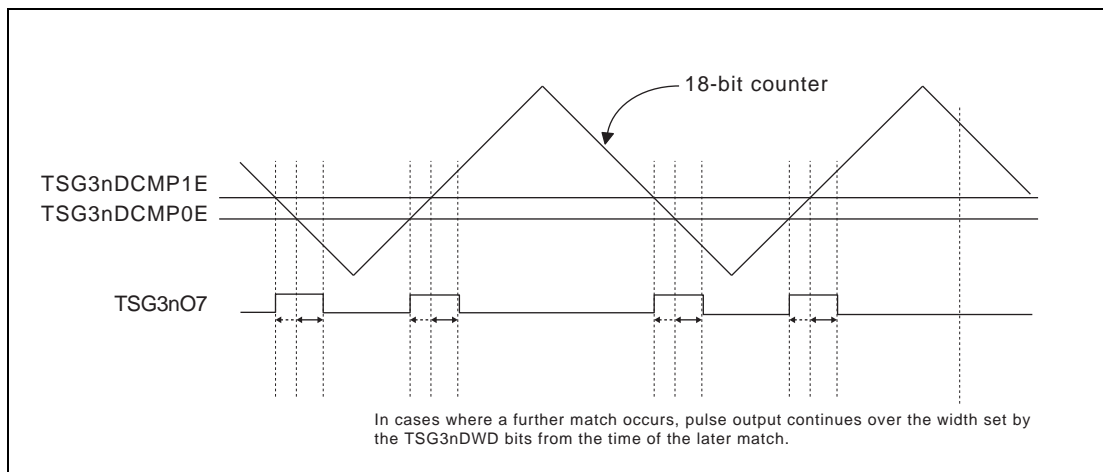


Figure 25.14 Example of TSG3nO7 Pin Diagnostic Pulse Output Timing (2) (with Pulse Output Width Overlapped)

25.4.1.5 Restart

Setting TSG3nTRG0.TSG3nTS to 1 or input to TSG3nTSST while the timer is operating (TSG3nSTR0.TSG3nTE = 1) leads to the following operations.

- Initializing the following counters;
 - 18-bit main counter, 18-bit sub-counter, interrupt skipping counter
- Initializing the following flags;
 - TSG3nCUF, TSG3nSUF, TSG3nRSF, and TSG3nTE of the TSG3nSTR0 register
 - TSG3nOPF2 to TSG3nOPF0 of the TSG3nSTR1 register (TSG3nTSF is not initialized)
 - TSG3nPPF, TSG3nPEF, TSG3nTDF, TSG3nNDF, TSG3nPRF, and TSG3nPTF of the TSG3nSTR2 register (TSG3nTBF2 to TSG3nTBF20 are not initialized)
- Reloading the following registers;
 - TSG3nCTL2, TSG3nCTL4, TSG3nIOC3, TSG3nCMP0, TSG3nCMP0E, TSG3nCMPmW (m = 1, 3, 5, 7, 9, or 11)
 - TSG3nCMP1 to TSG3nCMP12, TSG3nCMP1E to TSG3nCMP12E
 - TSG3nDCMP0W, TSG3nDCMP2, TSG3nDCMP0E to TSG3nDCMP2E
 - TSG3nPAT0W, TSG3nPAT1W
 - TSG3nCMPU, TSG3nCMPV, TSG3nCMPW
 - TSG3nCMPUE, TSG3nCMPVE, TSG3nCMPWE
 - TSG3nUPW, TSG3nVPW, TSG3nWPW
 - TSG3nUPWE, TSG3nVPWE, TSG3nWPWE
 - TSG3nHSPSHUE, TSG3nHSPSHVE, TSG3nHSPSHWE
 - TSG3nHSPCMUE, TSG3nHSPCMVE, TSG3nHSPCMWE
- Reflecting the settings in the following register;
 - TSG3nOPT0*¹

The dead time is always inserted because the dead time counter continues counting while the dead time control registers TSG3nDTC0W and TSG3nDTC1W are reloaded on restart.

If the counter is restarted while diagnostic output is in progress, the output is initialized.

Note 1. The settings in the TSG3nOPT0 register bits are reflected at the timings described below:

For TSG3nSOC

Switching from 0 to 1: immediately after written

Switching from 1 to 0: when the register is reloaded

For TSG3nSTE, TSG3nPOT, TSG3nPSS, TSG3nIDC, and TSG3nPSC

- At the next timer cycle
- When the output patterns are switched

25.4.2 Match Interrupt

The TSG3n can generate interrupts such as a compare match interrupt (INTTSG3nIm), a peak interrupt (INTTSG3nIPEK), and a valley interrupt (INTTSG3nIVLY). For an error interrupt and warning interrupt (INTTSG3nIER and INTTSG3nIWN), see **Section 25.4.6, Error/Warning Interrupt**.

A period interrupt (INTTSG3nI0) is generated for each timer period. In HT-PWM mode, it is generated when the TSG3nDTC0 buffer register value matches with the 18-bit counter value. When the 18-bit counter performs sawtooth waveform operation (PWM mode, SP-PWM mode, 120-DC mode, and HSP-PWM mode), it is generated after the 18-bit counter value has matched with the TSG3nCMP0E buffer register value.

A compare-match interrupt (INTTSG3nIm) is generated by a match of the TSG3nCMPmE buffer register value with the 18-bit counter value depending on the compare register to be used in each operating mode (m = 1 to 12).

A peak interrupt (INTTSG3nIPEK) is generated in all the modes. In HT-PWM mode, it is generated when the 18-bit counter switches from incrementing to decrementing. When the 18-bit counter performs sawtooth waveform operation (PWM mode, SP-PWM mode, 120-DC mode, and HSP-PWM mode), it is generated after the 18-bit counter value has matched with the TSG3nCMP0E buffer register value (the same timing as an INTTSG3nI0 interrupt).

A valley interrupt (INTTSG3nIVLY) is generated when the 18-bit counter switches from decrementing to incrementing in HT-PWM mode.

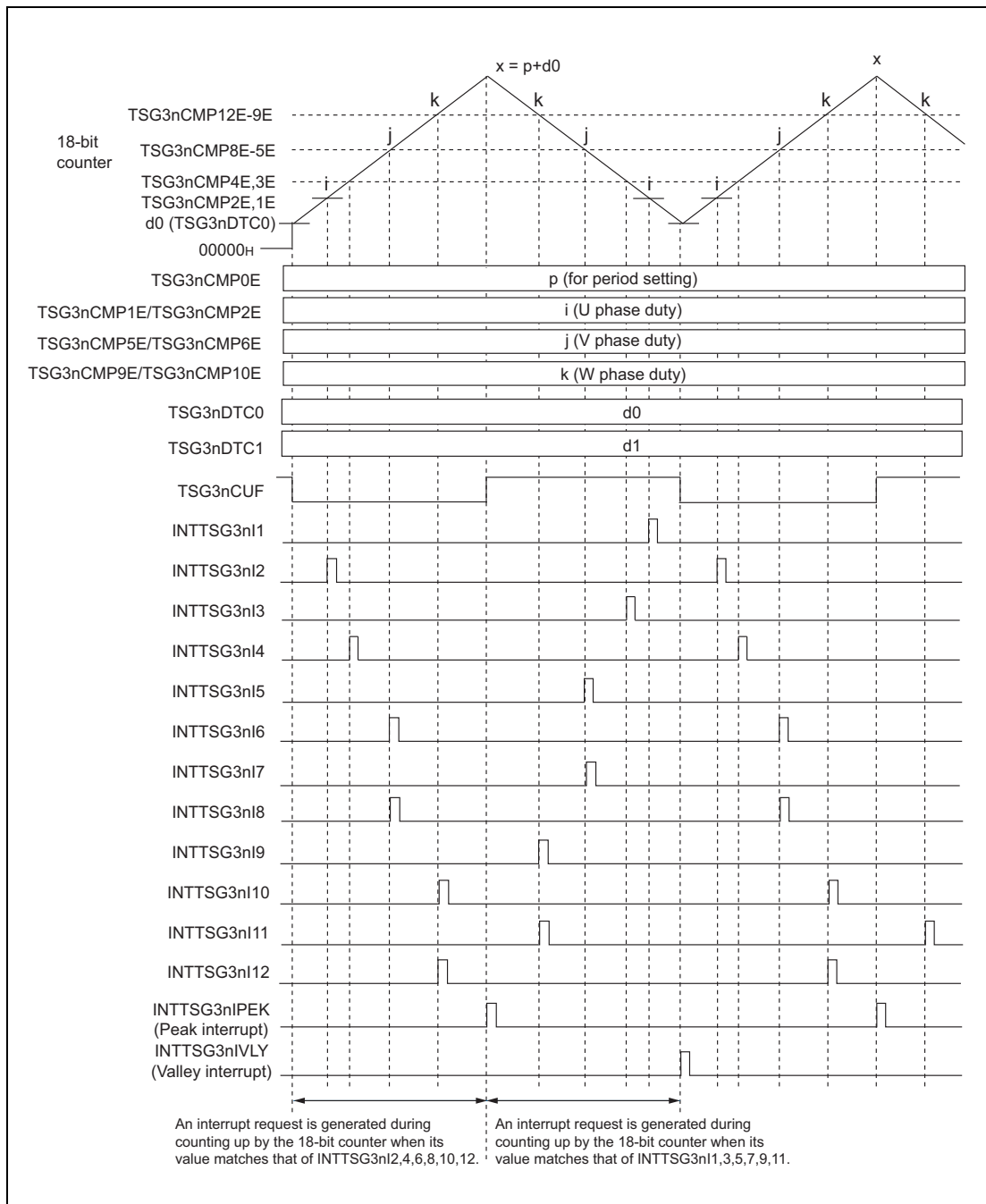


Figure 25.15 Interrupt Generation Example (Example of HT-PWM Mode)

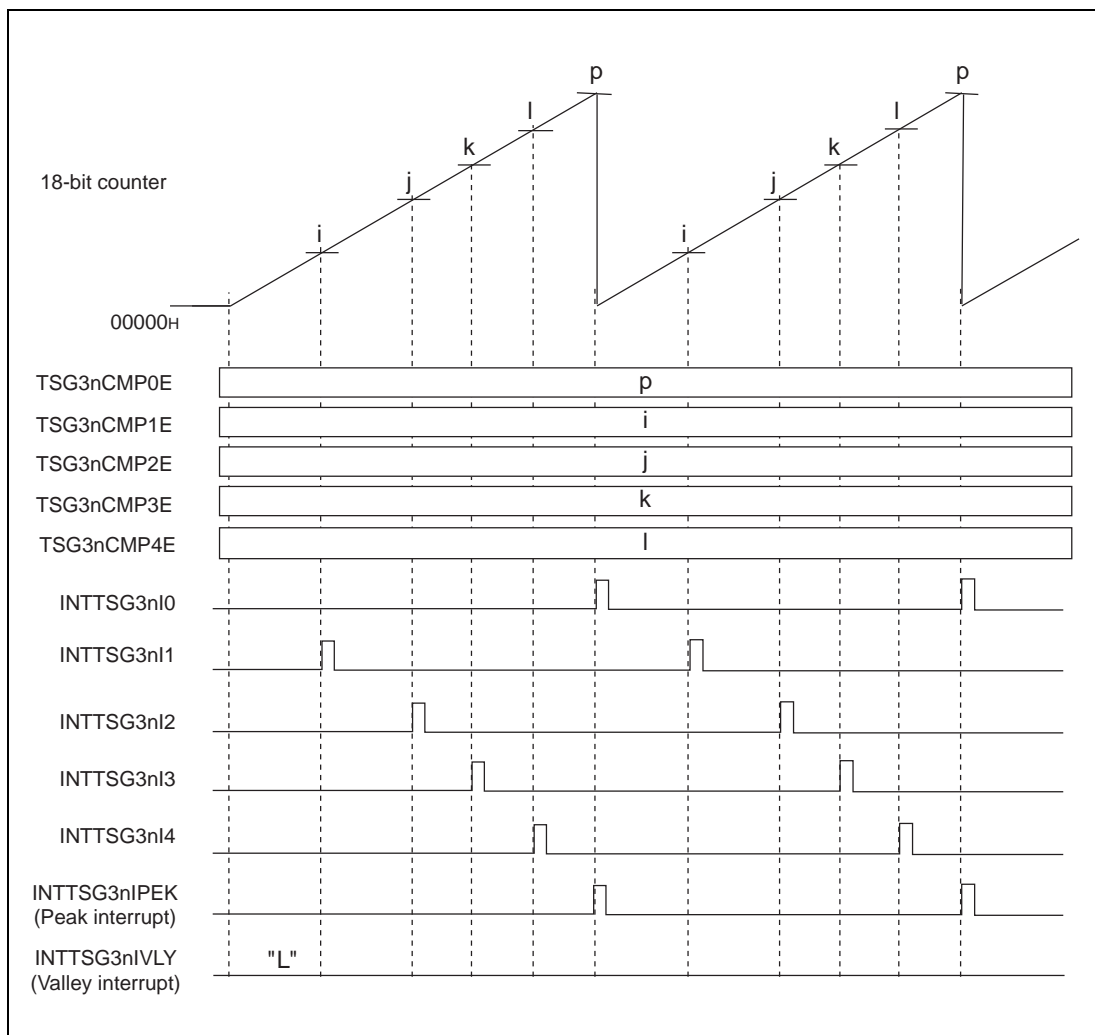


Figure 25.16 Interrupt Generation Example (Example of PWM Mode)

Interrupt in each mode (INTTSG3nI0-INTTSG3nI12, INTTSG3nIPEK, INTTSG3nIVLY, INTTSG3nIER, and INTTSG3nIWN) are listed in **Table 25.48**

Table 25.48 List of Interrupts in Each Mode (1/5)

| Operating Mode | INTTSG3nI0 | INTTSG3nI1 | INTTSG3nI2 | INTTSG3nI3 |
|----------------|------------------------------------|---|---|---|
| PWM mode | TSG3nCMP0E compare match interrupt | TSG3nCMP1E compare match interrupt*1 | TSG3nCMP2E compare match interrupt*1 | TSG3nCMP3E compare match interrupt*1 |
| HT-PWM mode | Period interrupt | TSG3nCMP1E compare match interrupt*2 when decrementing (TSG3nCUF = 1) | TSG3nCMP2E compare match interrupt*2 when incrementing (TSG3nCUF = 0) | TSG3nCMP3E compare match interrupt*2 when decrementing (TSG3nCUF = 1) |
| SP-PWM mode | TSG3nCMP0E compare match interrupt | TSG3nCMP1E compare match interrupt*1 | TSG3nCMP2E compare match interrupt*1 | — |
| 120-DC mode | TSG3nCMP0E compare match interrupt | TSG3nCMP1E compare match interrupt*1 | TSG3nCMP2E compare match interrupt*1 | TSG3nCMP3E compare match interrupt*1 |
| HSP-PWM mode | TSG3nCMP0E compare match interrupt | TSG3nCMP1E compare match interrupt*1 | TSG3nCMP2E compare match interrupt*1 | TSG3nCMP3E compare match interrupt*1 |

Note 1. A compare match interrupt is not generated when $TSG3nCMP0E < TSG3nCMPmE$ ($m = 1$ to 12).

Note 2. A compare match interrupt is not generated when $00000_H \leq TSG3nCMPmE < TSG3nDTC0$, $(TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE$.

Table 25.48 List of Interrupts in Each Mode (2/5)

| Operating Mode | INTTSG3nI4 | INTTSG3nI5 | INTTSG3nI6 | INTTSG3nI7 |
|----------------|---|---|---|---|
| PWM mode | TSG3nCMP4E compare match interrupt*1 | TSG3nCMP5E compare match interrupt*1 | TSG3nCMP6E compare match interrupt*1 | TSG3nCMP7E compare match interrupt*1 |
| HT-PWM mode | TSG3nCMP4E compare match interrupt*2 when incrementing (TSG3nCUF = 0) | TSG3nCMP5E compare match interrupt*2 when decrementing (TSG3nCUF = 1) | TSG3nCMP6E compare match interrupt*2 when incrementing (TSG3nCUF = 0) | TSG3nCMP7E compare match interrupt*2 when decrementing (TSG3nCUF = 1) |
| SP-PWM mode | — | TSG3nCMP5E compare match interrupt*1 | TSG3nCMP6E compare match interrupt*1 | — |
| 120-DC mode | TSG3nCMP4E compare match interrupt*1 | TSG3nCMP5E compare match interrupt*1 | TSG3nCMP6E compare match interrupt*1 | TSG3nCMP7E compare match interrupt*1 |
| HSP-PWM mode | TSG3nCMP4E compare match interrupt*1 | TSG3nCMP5E compare match interrupt*1 | TSG3nCMP6E compare match interrupt*1 | TSG3nCMP7E compare match interrupt*1 |

Note 1. A compare match interrupt is not generated when $TSG3nCMP0E < TSG3nCMPmE$ ($m = 1$ to 12).

Note 2. A compare match interrupt is not generated when $00000_H \leq TSG3nCMPmE < TSG3nDTC0$, $(TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE$.

Table 25.48 List of Interrupts in Each Mode (3/5)

| Operating Mode | INTTSG3nI8 | INTTSG3nI9 | INTTSG3nI10 | INTTSG3nI11 |
|----------------|---|---|--|--|
| PWM mode | TSG3nCMP8E compare match interrupt*1 | TSG3nCMP9E compare match interrupt*1 | TSG3nCMP10E compare match interrupt*1 | TSG3nCMP11E compare match interrupt*1 |
| HT-PWM mode | TSG3nCMP8E compare match interrupt*2 when incrementing (TSG3nCUF = 0) | TSG3nCMP9E compare match interrupt*2 when decrementing (TSG3nCUF = 1) | TSG3nCMP10E compare match interrupt*2 when incrementing (TSG3nCUF = 0) | TSG3nCMP11E compare match interrupt*2 when decrementing (TSG3nCUF = 1) |
| SP-PWM mode | — | TSG3nCMP9E compare match interrupt*1 | TSG3nCMP10E compare match interrupt*1 | — |
| 120-DC mode | TSG3nCMP8E compare match interrupt*1 | TSG3nCMP9E compare match interrupt*1 | TSG3nCMP10E compare match interrupt*1 | TSG3nCMP11E compare match interrupt*1 |
| HSP-PWM mode | TSG3nCMP8E compare match interrupt*1 | TSG3nCMP9E compare match interrupt*1 | TSG3nCMP10E compare match interrupt*1 | TSG3nCMP11E compare match interrupt*1 |

Note 1. A compare match interrupt is not generated when $TSG3nCMP0E < TSG3nCMPmE$ ($m = 1$ to 12).

Note 2. A compare match interrupt is not generated when $00000_H \leq TSG3nCMPmE < TSG3nDTC0$, $(TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE$.

Table 25.48 List of Interrupts in Each Mode (4/5)

| Operating Mode | INTTSG3nI12 | INTTSG3nIPEK | INTTSG3nIVLY |
|----------------|--|---|------------------|
| PWM mode | TSG3nCMP12E compare match interrupt* ¹ | Peak interrupt at the same timing with INTTSG3nI0 | — |
| HT-PWM mode | TSG3nCMP12E compare match interrupt* ² when incrementing (TSG3nCUF=0) | Peak interrupt | Valley interrupt |
| SP-PWM mode | — | Peak interrupt at the same timing with INTTSG3nI0 | — |
| 120-DC mode | TSG3nCMP12E compare match interrupt* ¹ | Peak interrupt at the same timing with INTTSG3nI0 | — |
| HSP-PWM mode | TSG3nCMP12E compare match interrupt* ¹ | Peak interrupt at the same timing with INTTSG3nI0 | — |

Note 1. A compare match interrupt is not generated when $TSG3nCMP0E < TSG3nCMPmE$ ($m = 1$ to 12).

Note 2. A compare match interrupt is not generated when $00000_H \leq TSG3nCMPmE < TSG3nDTC0$,
 $(TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE$.

Table 25.48 List of Interrupts in Each Mode (5/5)

| Operating Mode | INTTSG3nIER | INTTSG3nIWN |
|----------------|-----------------|-------------------|
| PWM mode | Error interrupt | Warning interrupt |
| HT-PWM mode | Error interrupt | Warning interrupt |
| SP-PWM mode | Error interrupt | Warning interrupt |
| 120-DC mode | Error interrupt | Warning interrupt |
| HSP-PWM mode | Error interrupt | Warning interrupt |

25.4.3 Flags

Table 25.49 List of Flags

| Number | Flag Name | Symbol | Registers | Operating Mode |
|--------|---|------------------------------|-----------|---------------------|
| (1) | Up count flag | TSG3nCUF | TSG3nSTR0 | HT-PWM mode |
| | | TSG3nSUF | TSG3nSTR0 | |
| (2) | Positive phase and inverse phase simultaneous active state detection flag | TSG3nTBF0- TSG3nTBF2 | TSG3nSTR2 | All operating modes |
| (3) | Reload request flag | TSG3nRSF | TSG3nSTR0 | All operating modes |
| (4) | Noise Detection Flag | TSG3nNDF | TSG3nSTR2 | All operating modes |
| (5) | Pattern order detection flag | TSG3nTSF | TSG3nSTR1 | All operating modes |
| (6) | Pattern error detection flag | TSG3nPEF | TSG3nSTR2 | All operating modes |
| (7) | Pattern reversal detection flag | TSG3nPRF | TSG3nSTR2 | All operating modes |
| (8) | TSG3nPTSI2 to TSG3nPTSI0 pin abnormal toggle detection flag | TSG3nPTF | TSG3nSTR2 | All operating modes |
| (9) | TSG3nOPCI0 and TSG3nOPCI1 signal simultaneous trigger detection flag | TSG3nTDF | TSG3nSTR2 | All operating modes |
| (10) | Pattern phase difference detection flag | TSG3nPPF | TSG3nSTR2 | All operating modes |
| (11) | Timer output pattern flag | TSG3nOPF0 to TSG3nOPF2 | TSG3nSTR1 | All operating modes |
| (12) | Pattern switch detection signal (internal signal) | TSG3nPTE | — | All operating modes |

25.4.3.1 Up Count Flag (TSG3nCUF and TSG3nSUF)

Name

Up count flag (TSG3nSTR0.TSG3nCUF and TSG3nSUF)

Description

There are following two up count flags.

TSG3nCUF is an up/down count flag of the 18-bit counter.

TSG3nSUF is an up/down count flag of the 18-bit sub-counter.

For both TSG3nCUF and TSG3nSUF, 0 means increment, and 1 means decrement.

TSG3nCUF and TSG3nSUF can be used only in HT-PWM mode.

Example of operation

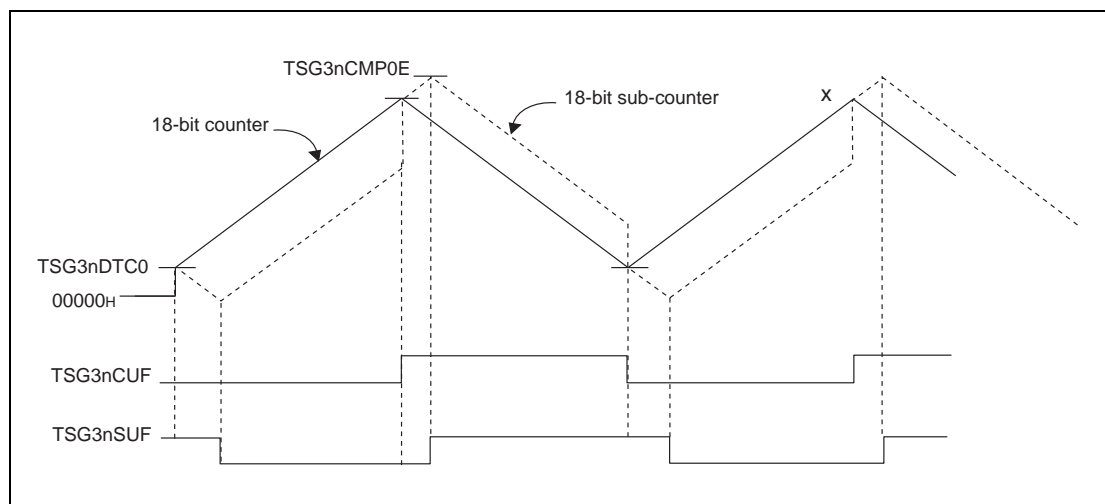


Figure 25.17 Example of Up count flag operation

NOTES

1. TSG3nCUF value is:
 - 0 (up count) when $TSG3nDTC0 \leq 18\text{-bit counter} \leq (TSG3nCMP0E + TSG3nDTC0 - 2)$
 - 1 (down count) when $(TSG3nCMP0E + TSG3nDTC0) \geq 18\text{-bit counter} \geq TSG3nDTC0 + 2$
2. TSG3nSUF value is:
 - 0 (up count) when $0 \leq 18\text{-bit sub-counter} \leq (TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 - 2)$
 - 1 (down count) when $(TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1) \geq 18\text{-bit sub-counter} \geq 2$

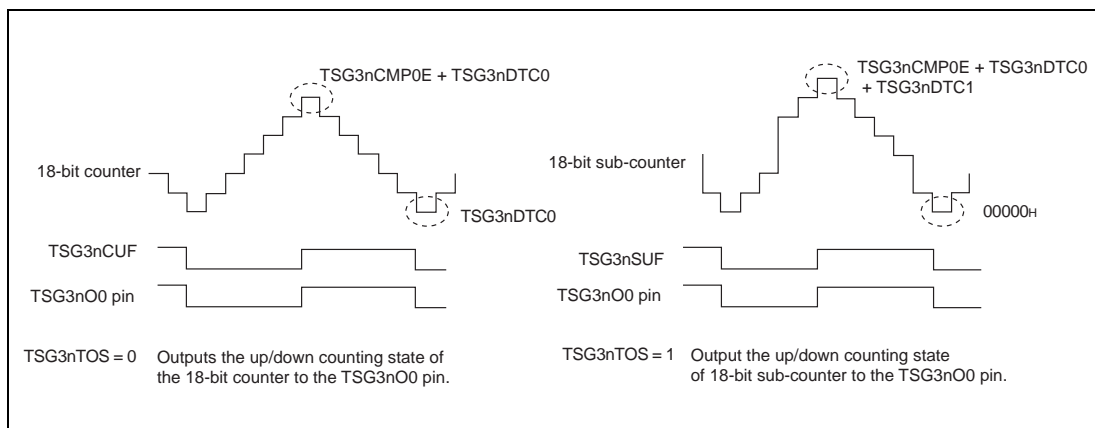


Figure 25.18 TSG3n00 Pin Output depending on TSG3nIOC1.TSG3nTOS Setting

Operating Mode

TSG3nCUF and TSG3nSUF can be used only in HT-PWM mode.

25.4.3.2 Positive Phase and Inverse Phase Simultaneous Active State Detection Flag (TSG3nTBF0 to TSG3nTBF2)

Name

Positive phase and inverse phase simultaneous active state detection flag (TSG3nSTR2.TSG3nTBF0 to TSG3nTBF2 flags)

Description

When any of TSG3nCTL1.TSG3nTBA2 to TSG3nTBA0 is 1, TSG3nTBF0 to TSG3nTBF2 can detect the simultaneous active state of the positive phase and inverse phase of TSG3n.

When the simultaneous active state of the positive phase and inverse phase of the TSG3n is detected, the corresponding TSG3nTBF0 to TSG3nTBF2 flags are set to 1, and an error interrupt (INTTSG3nIER) is generated. The flags are cleared when 1 is written to TSG3nSTC.TSG3nTBR0 to TSG3nTBR2, respectively.

Example of operation

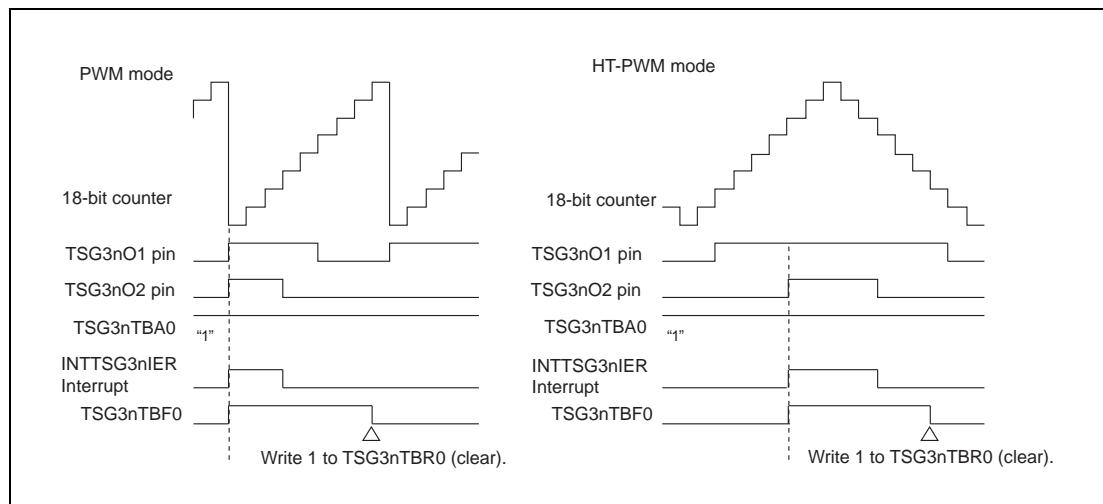


Figure 25.19 Example of Positive Phase and Inverse Phase Simultaneous Active State Detection Flag Operation

Operating Mode

Available in all operating modes.

CAUTION

TSG3nTBF0 to TSG3nTBF2 are valid only when TSG3nCTL1.TSG3nTBA0 to TSG3nTBA2 = 1 and TSG3nSTR0.TSG3nTE = 1.

25.4.3.3 Reload Request Flag (TSG3nRSF)

Name

Reload request flag (TSG3nSTR0.TSG3nRSF)

Description

TSG3nRSF is set to 1 when a reload request is generated (when a value is written to TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE)), and cleared to 0 when reload occurs and the value is transferred to all the buffer registers.

Example of operation

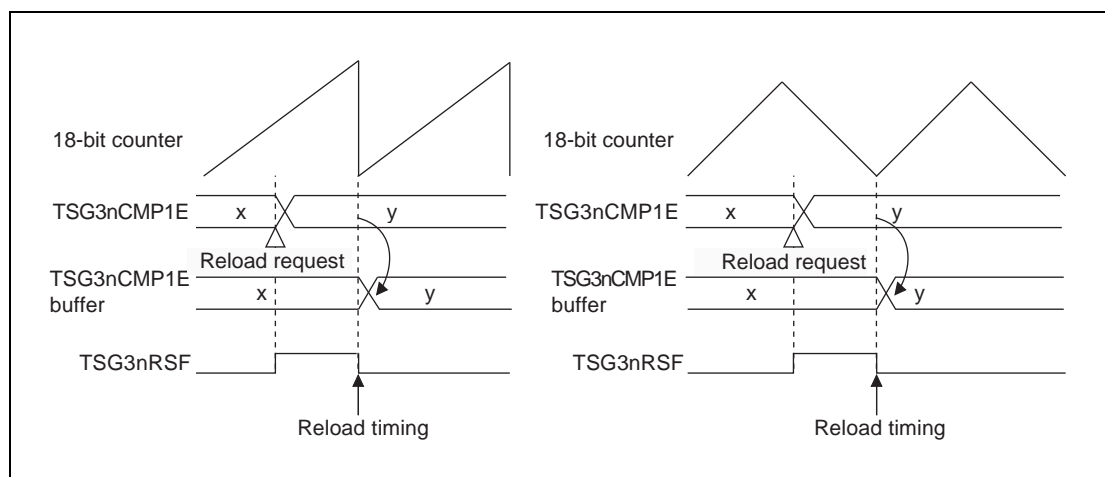


Figure 25.20 Example of Reload Request Flag Operation

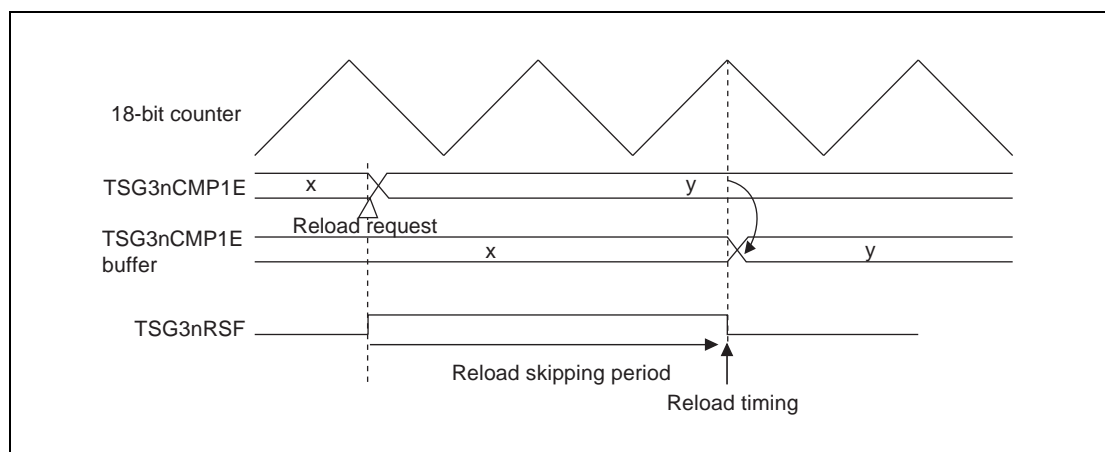


Figure 25.21 Reload Request Flag and Reload Skipping Period

Operating Mode

Available in all operating modes.

25.4.3.4 Noise Detection Flag (TSG3nNDF)

Name

Noise detection flag (TSG3nSTR2.TSG3nNDF)

Description

TSG3nNDF can detect that two or more pins of TSG3nPTSI2 to TSG3nPTSI0 have changed simultaneously (a noise is generated).

TSG3nNDF is set to 1 when two or more pins of TSG3nPTSI2 to TSG3nPTSI0 have changed simultaneously (a noise is generated), and a warning interrupt (INTTSG3nIWN) is generated. The TSG3nNDF flag is cleared to 0 when 1 is written to the TSG3nSTC.TSG3nNDR bit.

Example of operation

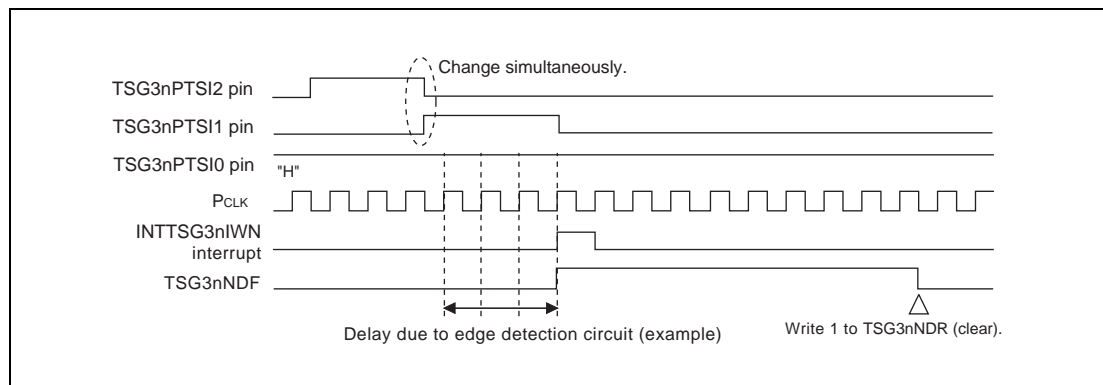


Figure 25.22 Example of Noise Detection Flag Operation

Operation mode

Available in all operating modes.

CAUTION

TSG3nNDF is valid only when TSG3nCTL1.TSG3nNDC = 1 and TSG3nSTR0.TSG3nTE = 1.

25.4.3.5 Pattern Order Detection Flag (TSG3nTSF)

Name

Pattern order detection flag (TSG3nSTR1.TSG3nTSF)

Description

TSG3nTSF can detect the order of patterns input to the TSG3nPTSI2 to TSG3nPTSI0 pins.

TSG3nTSF is set depending on the values input to the TSG3nPTSI2 to TSG3nPTSI0 pins as shown in the table below

Table 25.50 Pattern Order Detection Flag and Pattern Input Order

| TSG3nTSF | Values input to TSG3nPTSI2 to TSG3nPTSI0 Pins |
|----------|---|
| 0 | [1,0,1] → [1,0,0] → [1,1,0] → [0,1,0] → [0,1,1] → [0,0,1] |
| 1 | [1,0,1] ← [1,0,0] ← [1,1,0] ← [0,1,0] ← [0,1,1] ← [0,0,1] |

Example of operation

(a) When Normal Input to TSG3nPTSI2 to TSG3nPTSI0 Pins is Detected

As shown in **Figure 25.23**, if the TSG3nPTSI2 to TSG3nPTSI0 pins change in the normal order, 0 or 1 is set according to the change order at the change timing.

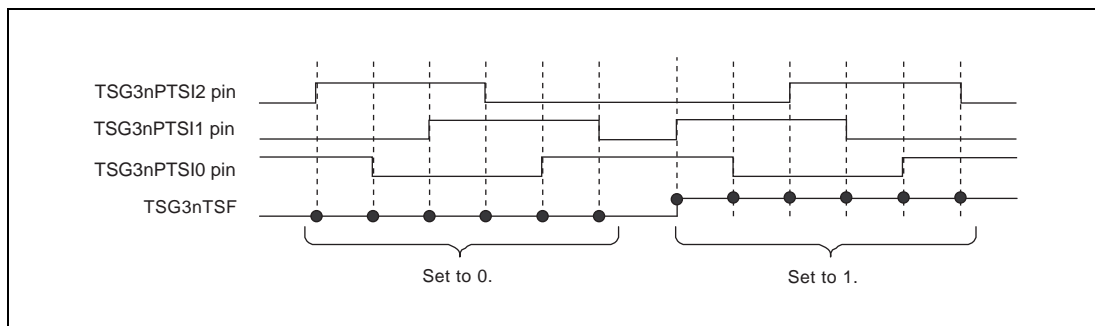


Figure 25.23 Example of Pattern Order Detection Flag Operation (Normal Operation)

(b) Detection of Change in Input Pattern Order

Immediately after TSG3n starts operation, the rotation direction cannot be determined. Therefore, TSG3nTSF cannot detect the change (normal or reverse rotation) in the patterns input to the TSG3nPTS12 to TSG3nPTS10 pins. To enable detection of change immediately after the beginning of operation, TSG3nPSC should be set before operation starts (when TSG3nTE = 0, the TSG3nPSC value is reflected).

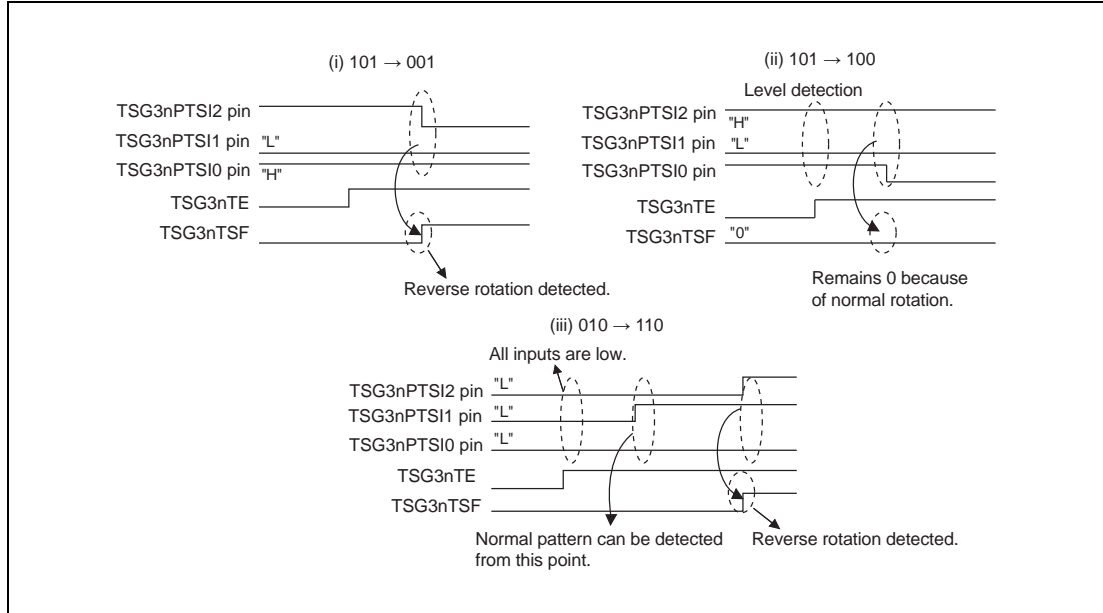


Figure 25.24 Example of Detecting Change (Normal/Reverse Rotation) in Pattern Input to TSG3nPTS12 to TSG3nPTS10 Pins

(c) When Abnormal Input to TSG3nPTS12 to TSG3nPTS10 Pins is Detected

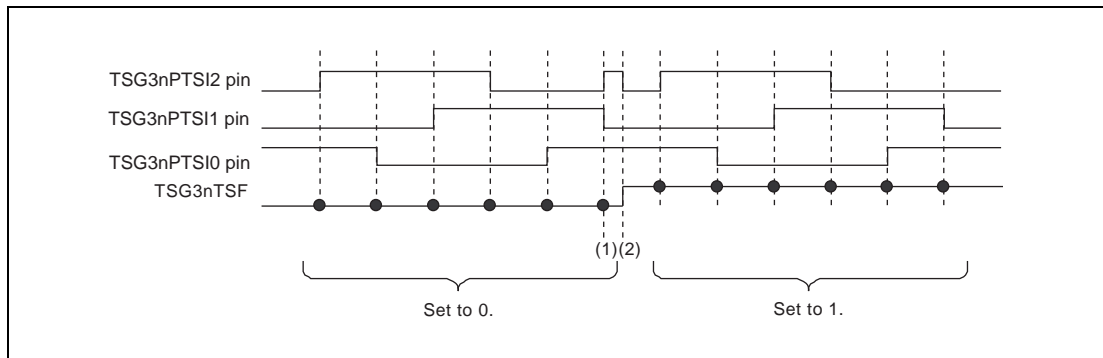


Figure 25.25 Example of Operation when Values Input to Two Pins of TSG3nPTS12 to TSG3nPTS10 Change (Abnormal Operation)

- (1) TSG3nTSF does not change at this point because it expects the input pattern change to {0, 1, 0} or {0, 0, 1} (if values of two pins change, TSG3nTSF does not change).
- (2) TSG3nPTS12 to TSG3nPTS10 pins are determined to have been changed from {1, 0, 1} to {0, 0, 1}, and TSG3nTSF is set to 1.

Operation mode

Available in all operating modes.

25.4.3.6 Pattern Error Detection Flag (TSG3nPEF)

Name

Pattern error detection flag (TSG3nSTR2.TSG3nPEF)

Description

TSG3nPEF can detect that 000 or 111 is input to the TSG3nPTSI2 to TSG3nPTSI0 pins.

TSG3nPEF is set to 1 when the levels of the TSG3nPTSI2 to TSG3nPTSI0 pins are 111 or 000, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPEF is cleared to 0 when 1 is written to TSG3nSTC.TSG3nPER.

Example of operation

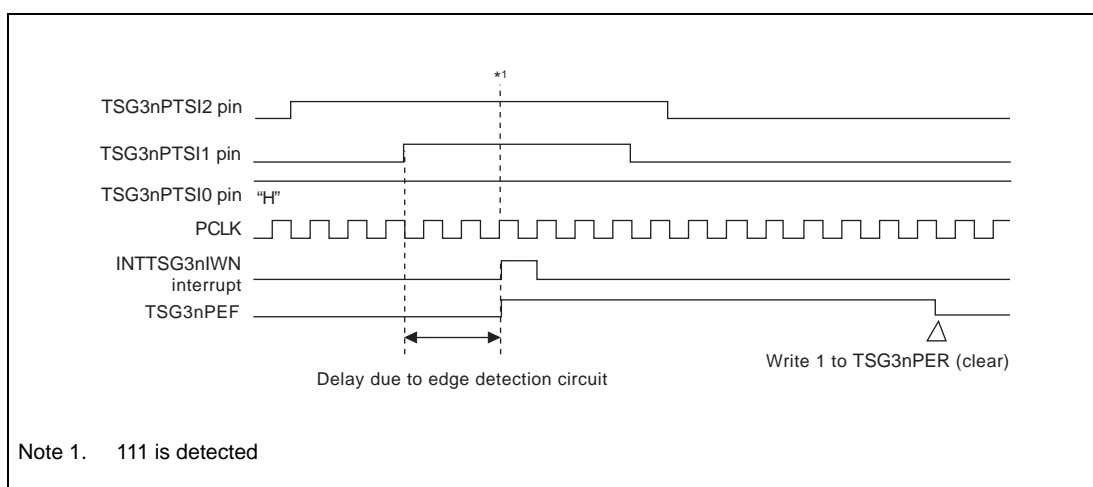


Figure 25.26 Example of Pattern Error Detection Flag Operation (TSG3nPTSI2 to TSG3nPTSI0 Pins = 111)

Operation mode

Available in all operating modes.

CAUTIONS

TSG3nPEF is valid only when TSG3nCTL1.TSG3nPEC = 1 and TSG3nSTR0.TSG3nTE = 1.

25.4.3.7 Pattern Reversal Detection Flag (TSG3nPRF)

Name

Pattern reversal detection flag (TSG3nSTR2.TSG3nPRF)

Description

TSG3nPRF can detect that the pattern change order of the TSG3nPTSI2 to TSG3nPTSI0 pins have been reversed.

TSG3nPRF is set to 1 when the pattern order detection flag (TSG3nTSF) changes, and a warning interrupt (INTTSG3nIWN) is generated. However, immediately after TSG3nSTR0.TSG3nTE is set to 1, TSG3nPRF becomes valid at the timing of the second and subsequent change in TSG3nPTSI2 to TSG3nPTSI0 pins.

TSG3nPRF is cleared to 0 when 1 is written to the TSG3nSTC.TSG3nPRR bit.

Example of operation

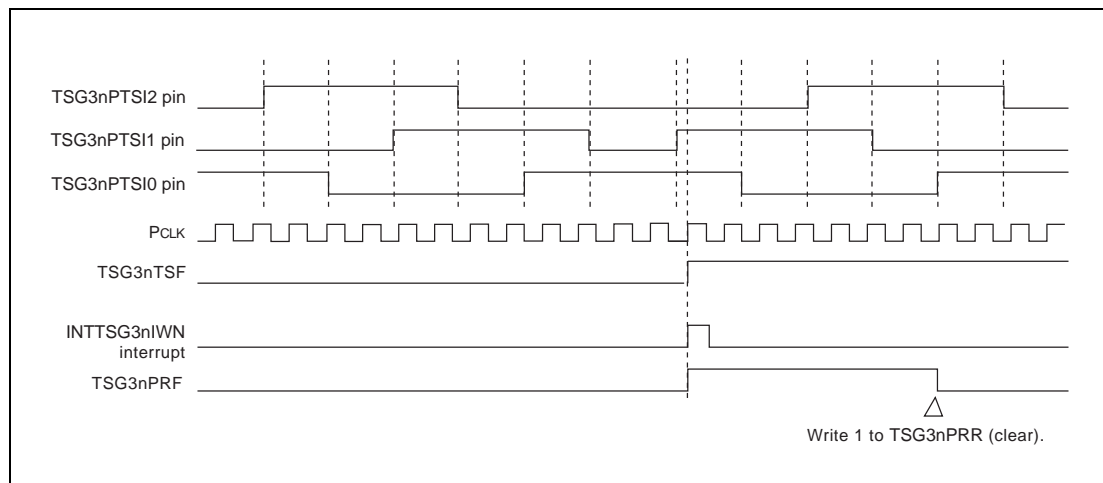


Figure 25.27 Example of Pattern Reversal Detection Flag Operation

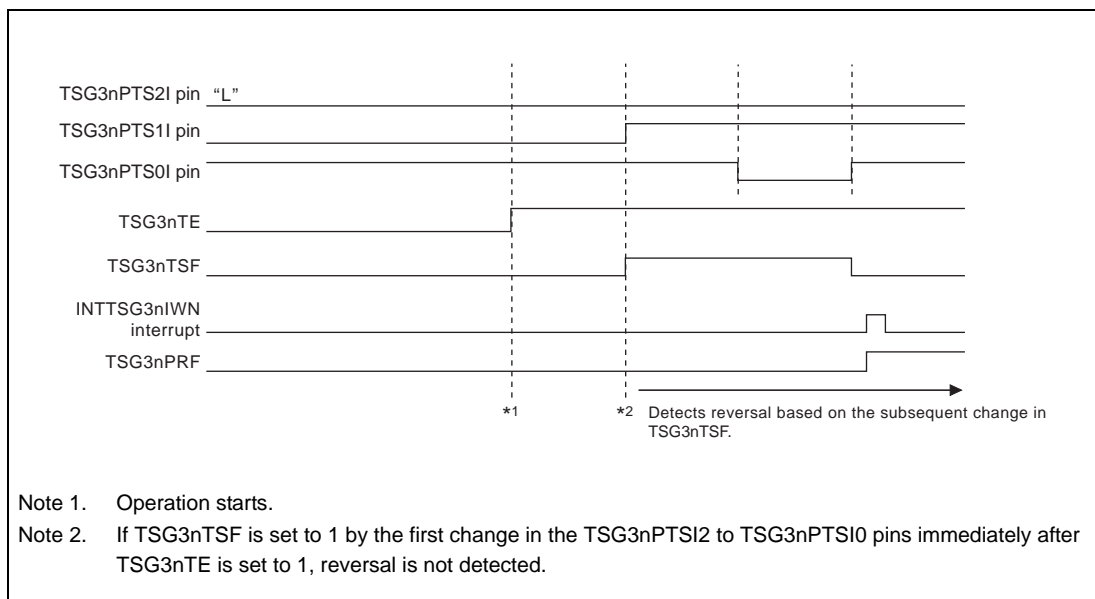


Figure 25.28 Example of Operation immediately after TSG3nTE Flag in TSG3nSTR0 is Set to 1

Operation mode

Available in all operating modes.

CAUTION

TSG3nPRF is valid only when TSG3nCTL1.TSG3nPRC = 1 and TSG3nSTR0.TSG3nTE = 1.

25.4.3.8 TSG3nPTSI2 to TSG3nPTSI0 Pin Abnormal Toggle Detection Flag (TSG3nPTF)

Name

TSG3nPTSI2 to TSG3nPTSI0 pin abnormal toggle detection flag (TSG3nSTR2.TSG3nPTF)

Description

TSG3nPTF can detect that the values of the TSG3nPTSI2 to TSG3nPTSI0 pins change three or more times during the TSG3nOPCI0 or TSG3nOPCI1 signal trigger.

TSG3nPTF is set to 1 when the third trigger of TSG3nOPCI0 or TSG3nOPCI1 signal occurs simultaneously with the change in TSG3nPTSI2 to TSG3nPTSI0, and a warning interrupt (INTTSG3nIWN) is generated.

TSG3nPTF is cleared to 0 when 1 is written to TSG3nSTC.TSG3nPTR.

Example of operation

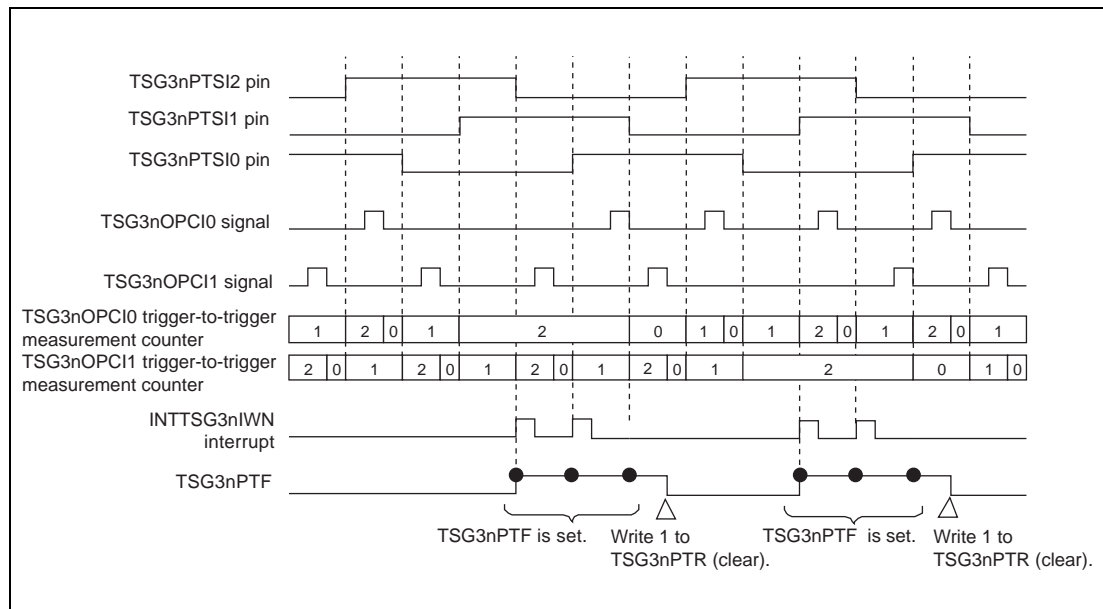


Figure 25.29 Example of TSG3nPTSI2 to TSG3nPTSI0 Pin Abnormal Toggle Detection Flag Operation

Operating mode

Available in all operating modes.

NOTES

1. TSG3nPTF is valid only when TSG3nCTL1.TSG3nPTC1 bit = 1 and TSG3nSTR0.TSG3nTE = 1.
2. When TSG3nPTC0 bit = 1 and TSG3nPTC1 bit = 1, TSG3nO1 to TSG3nO6 pin output switch control is automatically switched to pattern switch method (TSG3nOPT0.TSG3nPOT bit = 0) if an abnormal toggle is detected.

25.4.3.9 TSG3nOPCI0 and TSG3nOPCI1 Signal Simultaneous Trigger Detection Flag (TSG3nTDF)

Name

TSG3nOPCI0 and TSG3nOPCI1 signal simultaneous trigger detection flag (TSG3nSTR2.TSG3nTDF)

Description

TSG3nTDF can detect that TSG3nOPCI0 and TSG3nOPCI1 signals are generated simultaneously.

TSG3nTDF is set to 1 when the TSG3nOPCI0 and TSG3nOPCI1 signals are generated simultaneously, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nTDF is cleared to 0 when 1 is written to TSG3nSTC.TSG3nTDR.

Example of operation

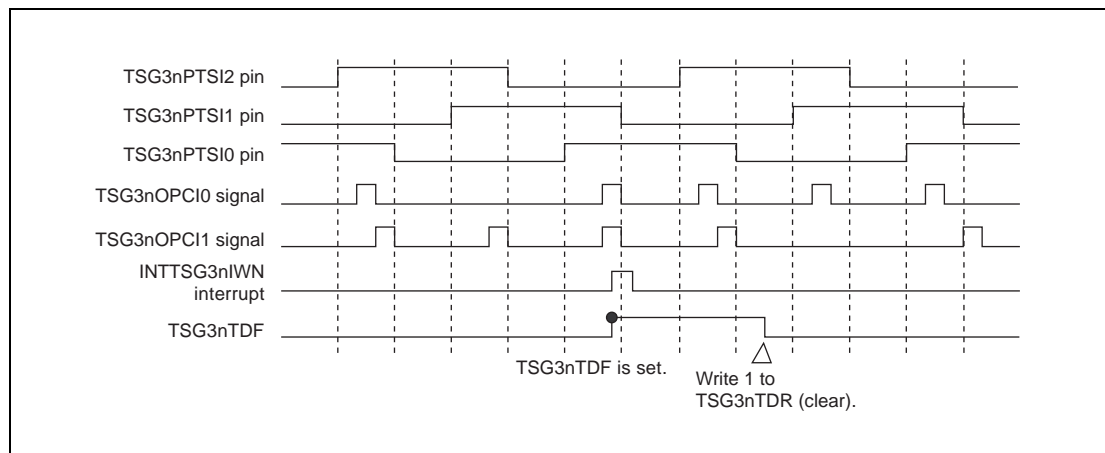


Figure 25.30 Example of TSG3nOPCI0 and TSG3nOPCI1 Signal Simultaneous Trigger Detection Flag Operation

Operating mode

Available in all operating modes.

CAUTION

TSG3nTDF is valid only when TSG3nCTL1.TSG3nTDC = 1 and TSG3nSTR0.TSG3nTE = 1.

25.4.3.10 Pattern Phase Difference Detection Flag (TSG3nPPF)

Name

Pattern phase difference detection flag (TSG3nSTR2.TSG3nPPF)

Description

TSG3nPPF can detect the phase difference between the input pattern (TSG3nPTS12 to TSG3nPTS10 pins) and the output pattern (TSG3nSTR1.TSG3nOPF2 to TSG3nOPF0 flags).

TSG3nPPF is set to 1 when the pattern phase difference is detected when the TSG3nOPCI0 and TSG3nOPCI1 signal triggers are input, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPPF remains 1 until it is cleared to 0 when 1 is written to TSG3nSTC.TSG3nPPR by software. When the phase difference is detected, TSG3nPPF is set at each operation clock cycle (PCLK). TSG3nPPF should be cleared to 0 when no phase difference occurs.

Table 25.51 Correspondence between Normal Input Patterns and Output Patterns

| | | | | | | |
|---------------------------------------|---------|---------|---------|---------|---------|---------|
| TSG3nPTS12 to TSG3nPTS10 pins (input) | "1,0,1" | "1,0,0" | "1,1,0" | "0,1,0" | "0,1,1" | "0,0,1" |
| TSG3nOPF2 to TSG3nOPF0 flags (output) | "0,0,1" | "1,0,1" | "1,0,0" | "1,1,0" | "0,1,0" | "0,1,1" |
| | "1,0,1" | "1,0,0" | "1,1,0" | "0,1,0" | "0,1,1" | "0,0,1" |
| | "1,0,0" | "1,1,0" | "0,1,0" | "0,1,1" | "0,0,1" | "1,0,1" |

Example of operation

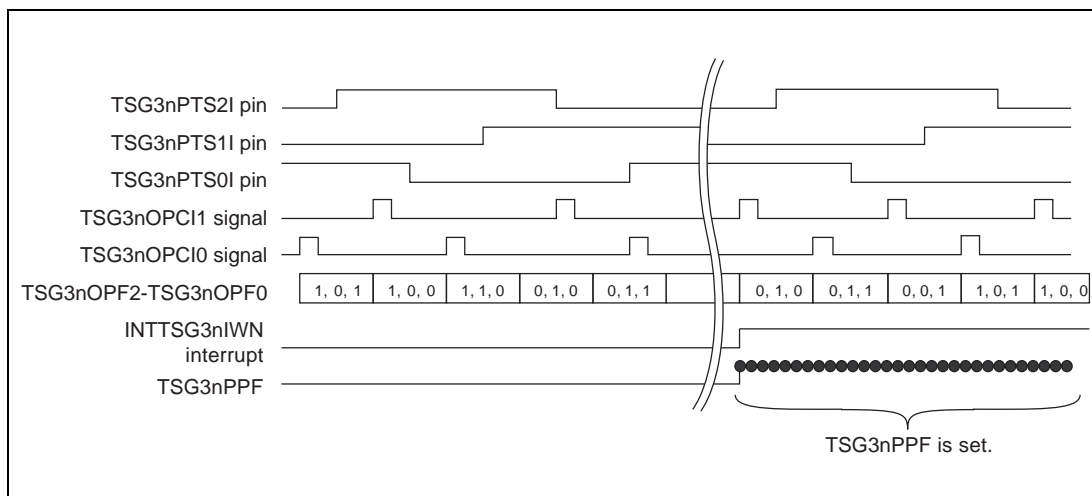


Figure 25.31 Example of Pattern Difference Detection Flag Operation

Operating mode

Available in all operating modes.

CAUTIONS

1. TSG3nPPF is valid only when TSG3nCTL1.TSG3nPPC = 1 and TSG3nSTR0.TSG3nTE = 1.
2. When 000 or 111 is input to the TSG3nPTS12 to TSG3nPTS10 pins, or when TSG3nOPF2 to TSG3nOPF0 are set to 000 or 111, TSG3nPPF is not set.

25.4.3.11 Timer Output Pattern Flag (TSG3nOPF2-TSG3nOPF0)

Name

Timer output pattern flag (TSG3nSTR1.TSG3nOPF2 to TSG3nOPF0)

Description

TSG3nOPF2 to TSG3nOPF0 flags indicate the timer output patterns.

For details, see **Section 25.4.7.4, 120-DC Mode**, and **Section 25.4.7.8, Software Output Control Function**.

Operating mode

Available in all operating modes.

25.4.3.12 Pattern Switch Detection Signal (TSG3nPTE)

Name

Pattern switch detection signal (TSG3nPTE signal)

Description

The TSG3nPTE signal toggles when the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes.

The toggle pattern is determined by the TSG3nPSC bit (TSG3nOPT0.TSG3nPSS = 1).

Table 25.52 Change Timing of Pattern Switch Detection Signal (1/2)

- TSG3nPSC = 0

| | | TSG3nPTSI2-TSG3nPTSI0 Pins after Change | | | | | | | |
|--|-----|---|-----|--------|--------|--------|--------|--------|--------|
| | | 000 | 111 | 101 | 100 | 110 | 010 | 011 | 001 |
| Current TSG3nPTSI2 to TSG3nPTSI0 pins | 000 | — | — | — | — | — | — | — | — |
| | 111 | — | — | — | — | — | — | — | — |
| | 101 | — | — | — | Toggle | — | — | — | — |
| | 100 | — | — | — | — | Toggle | — | — | — |
| | 110 | — | — | — | — | — | Toggle | — | — |
| | 010 | — | — | — | — | — | — | Toggle | — |
| | 011 | — | — | — | — | — | — | — | Toggle |
| | 001 | — | — | Toggle | — | — | — | — | — |

Table 25.52 Change Timing of Pattern Switch Detection Signal (2/2)

- TSG3nPSC = 1

| | | TSG3nPTSI2-TSG3nPTSI0 Pins after Change | | | | | | | |
|--|-----|---|-----|--------|--------|--------|--------|--------|--------|
| | | 000 | 111 | 101 | 100 | 110 | 010 | 011 | 001 |
| Current TSG3nPTSI2 to TSG3nPTSI0 pins | 000 | — | — | — | — | — | — | — | — |
| | 111 | — | — | — | — | — | — | — | — |
| | 101 | — | — | — | — | — | — | — | Toggle |
| | 100 | — | — | Toggle | — | — | — | — | — |
| | 110 | — | — | — | Toggle | — | — | — | — |
| | 010 | — | — | — | — | Toggle | — | — | — |
| | 011 | — | — | — | — | — | Toggle | — | — |
| | 001 | — | — | — | — | — | — | Toggle | — |

Example of operation

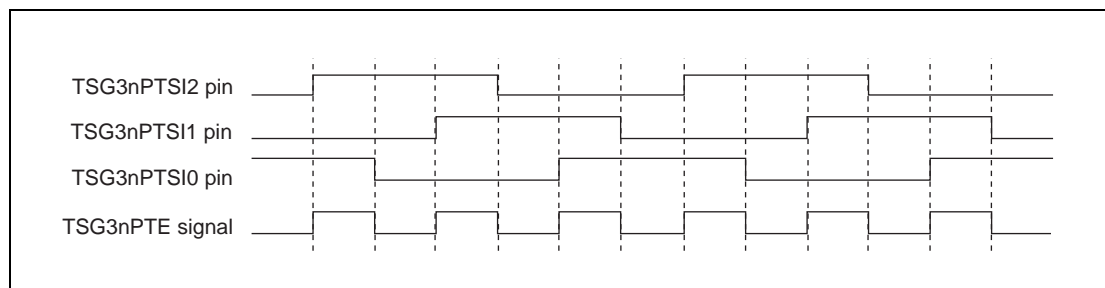


Figure 25.32 Example of Pattern Switch Detection Signal Operation

Operating mode

Available in all operating modes.

CAUTION

The TSG3nPTE signal is valid only when TSG3nIOC1.TSG3nPTS = 1 and TSG3nSTR0.TSG3nTE = 1.

25.4.4 Interrupt Skipping Function

Operation related to the interrupt skipping function is described below.

- Peak interrupts (INTTSG3nIPEK) and valley interrupts (INTTSG3nIVLY) can be skipped.
- TSG3nCTL4.TSG3nPIE enables outputting of the INTTSG3nIPEK interrupt and specifies whether to skip the interrupts.
- TSG3nCTL4.TSG3nVIE enables outputting of the INTTSG3nIVLY interrupt and specifies whether to skip the interrupts.

When TSG3nCTL3.TSG3nRIA is set to 1 (with reload skipping), reload is executed at the same timing as the interrupt after being skipped.

When TSG3nCTL3.TSG3nRIA is set to 0 (without reload skipping), reload is executed at the specified reload timing regardless of interrupt skipping.

CAUTION

When a value is written to TSG3nCTL4, and TSG3nRCC04 to TSG3nRCC00 are transferred to the buffer register, the interrupt skipping counter is cleared. Therefore, when the interrupt skipping function is used, interrupt interval may temporarily become longer. To avoid this, the interrupt skipping count should be changed with the reload timing being set to the interrupts skipped (TSG3nCTL3.TSG3nRIA = 1).

25.4.4.1 Operation of Interrupt Skipping Function

Timing diagram of interrupt skipping function in various conditions are shown in the following figures.

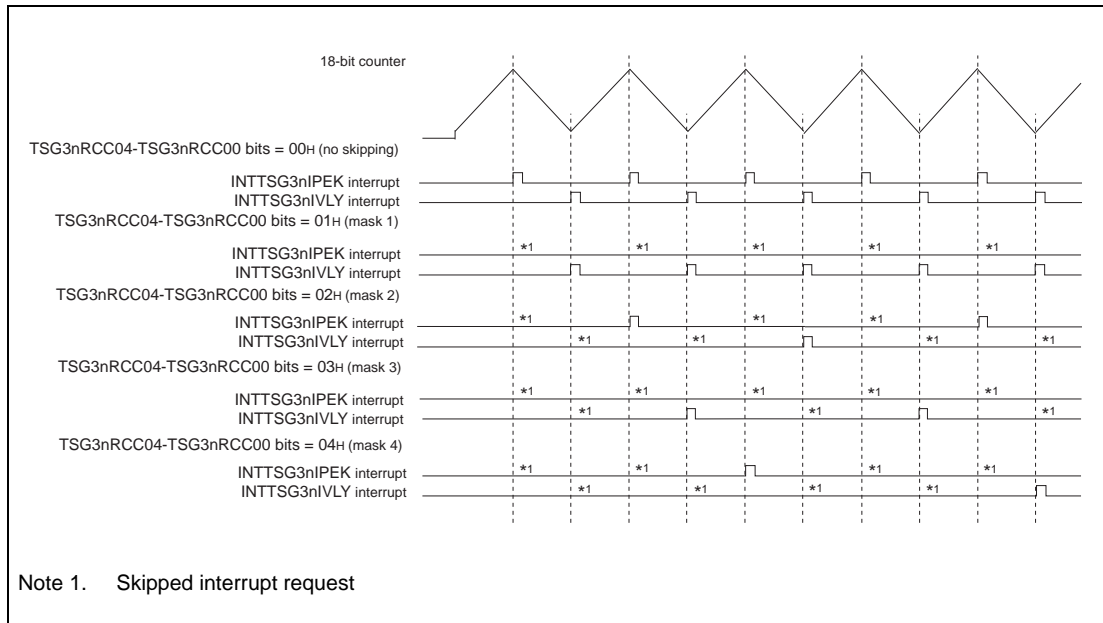


Figure 25.33 Interrupt Skipping Operation when TSG3nPIE = 1 and TSG3nVIE = 1 in TSG3nCTL4 Register (Peak and Valley Interrupt Generation in HT-PWM Mode)

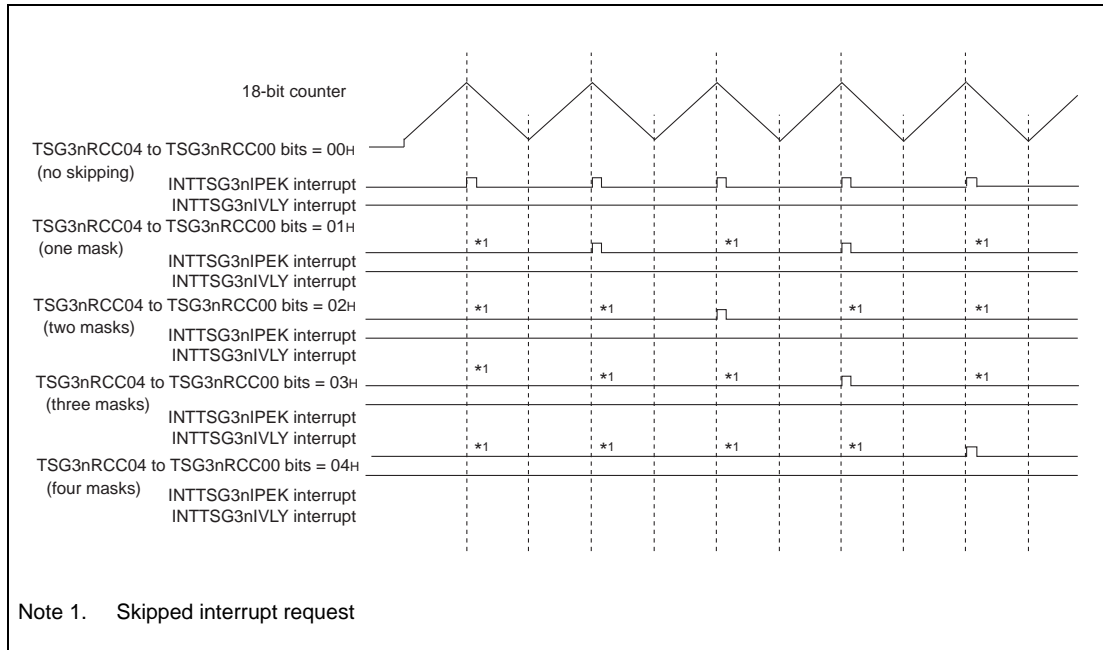


Figure 25.34 Interrupt Skipping Operation when TSG3nPIE = 1 and TSG3nVIE = 0 in TSG3nCTL4 Register (only Peak Interrupt Generation in HT-PWM Mode)

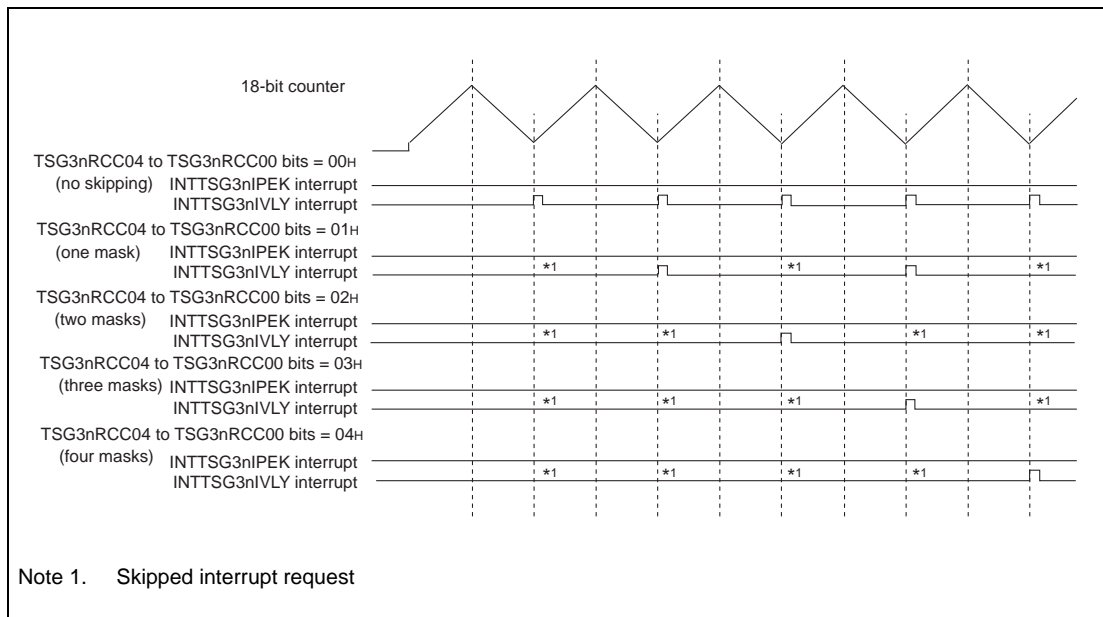


Figure 25.35 Interrupt Skipping Operation when TSG3nPIE = 0 and TSG3nVIE = 1 in TSG3nCTL4 Register (only Valley Interrupt Generation in HT-PWM Mode)

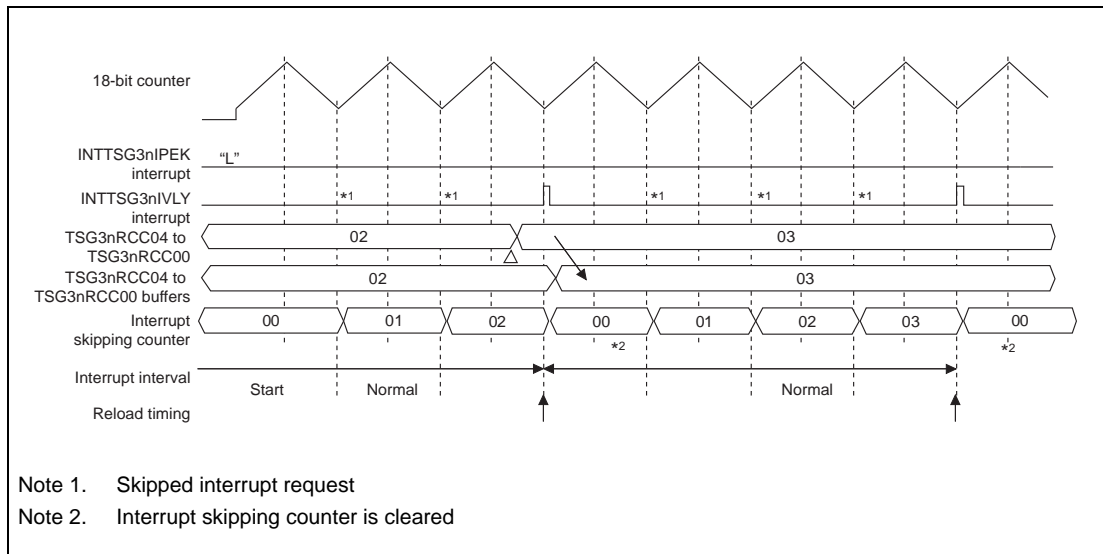


Figure 25.36 When TSG3nRMC = 0, TSG3nRIA = 1 in TSG3nCTL3 Register (with Reload Skipping)

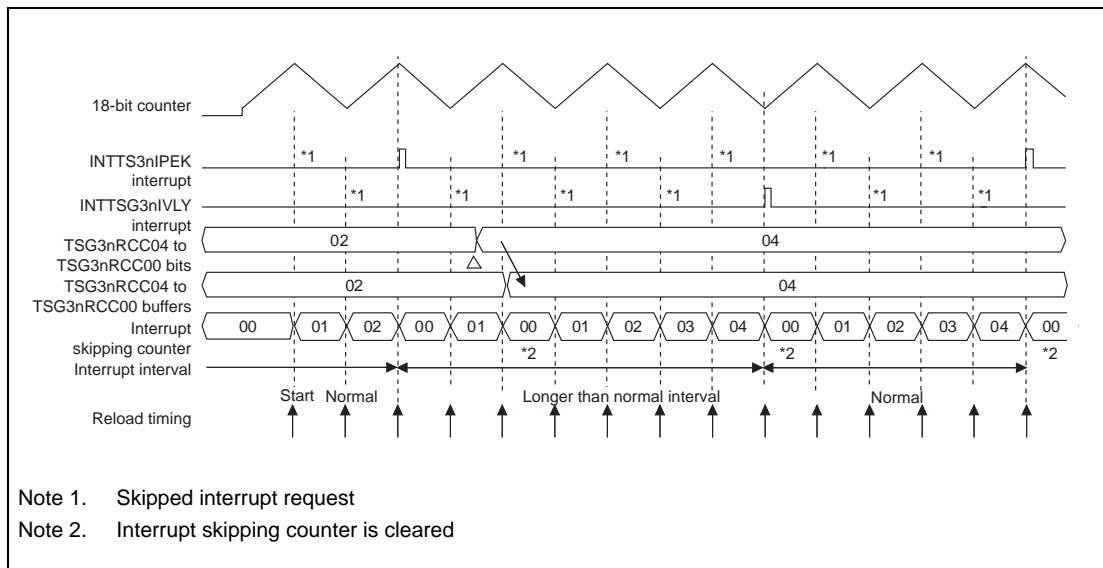


Figure 25.37 When TSG3nRMC = 0, TSG3nRIA = 0 in TSG3nCTL3 Register (without Reload Skipping)

CAUTION

Interrupt interval might be longer.

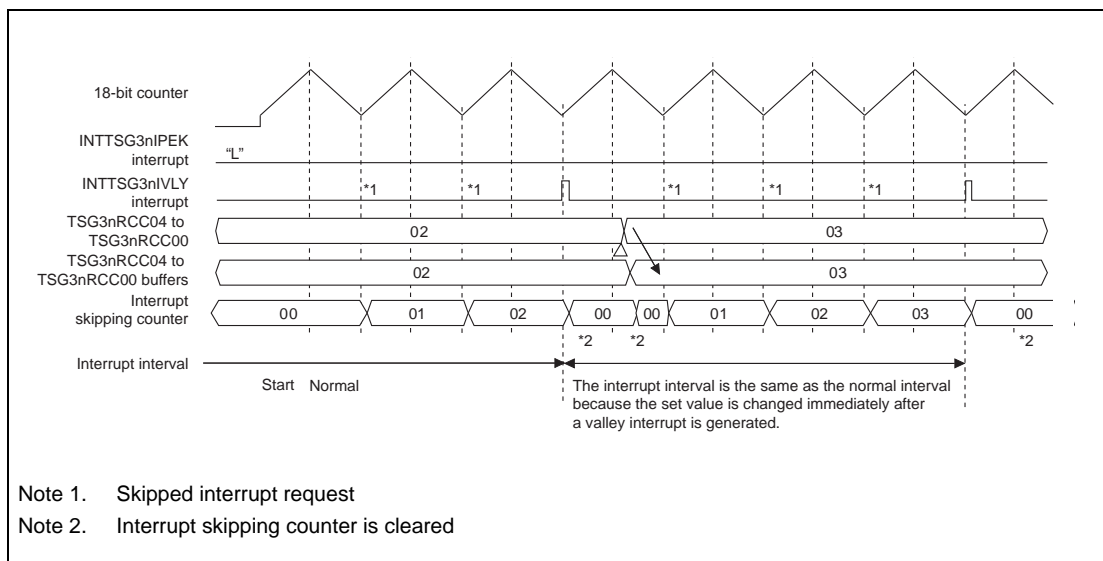


Figure 25.38 When TSG3nRMC = 1 in TSG3nCTL3 Register (Anytime Rewrite Mode)

NOTE

After rewriting, the value is reflected immediately regardless of the reload timing. The interrupt skipping counter is cleared when the value is transferred to the TSG3nRCC04 to TSG3nRCC00 buffers, not when the pertinent register is rewritten.

25.4.4.2 Example of Operation when Peak Interrupt is Generated (in PWM Mode)

Operation related to the interrupt skipping function in PWM mode is described below.

- Peak interrupts (INTTSG3nIPEK) can be skipped. In PWM mode, it is generated by compare match of TSG3nCMP0E buffer register and 18-bit counter.
- TSG3nCTL4.TSG3nPIE enables outputting of the INTTSG3nIPEK interrupt and specifies whether to skip the interrupts.
- The setting of TSG3nCTL4.TSG3nVIE is disabled. At this time, the INTTSG3nIVLY interrupt is not generated.

When TSG3nCTL3.TSG3nRIA is set to 1 (with reload skipping), reload is executed at the same timing as the interrupt after being skipped.

CAUTION

When a value is written to TSG3nCTL4, and TSG3nRCC04 to TSG3nRCC00 are transferred to the buffer register, the interrupt skipping counter is cleared. Therefore, when the interrupt skipping function is used, interrupt interval may be long temporarily. To avoid this, the interrupt skipping count should be changed with the reload timing being set to the interrupts skipped (TSG3nCTL3.TSG3nRIA = 1).

(1) Example of operation

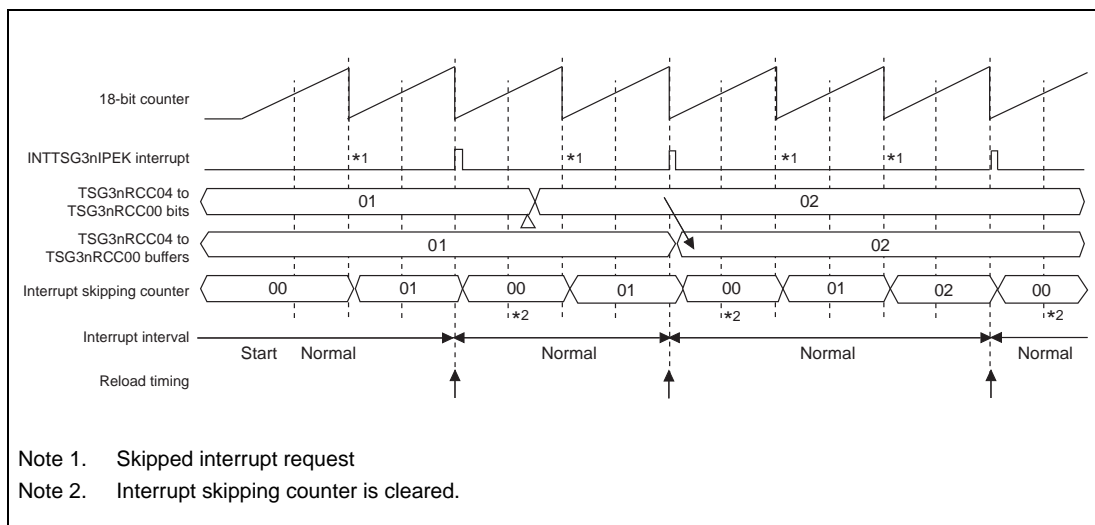


Figure 25.39 When TSG3nCTL3.TSG3nRMC = 0, TSG3nRIA = 1, TSG3nCTL4.TSG3nPRE = 1 (Recommended Setting)

NOTE

When TSG3nCTL3.TSG3nRIA = 1, reload is executed at the same timing as the interrupt after being skipped.

25.4.5 A/D Conversion Trigger Function

A/D conversion trigger operation is described below.

The TSG3nDCMP0E, TSG3nDCMP1E, and TSG3nDCMP2E registers are used as compare registers of the A/D conversion trigger function.

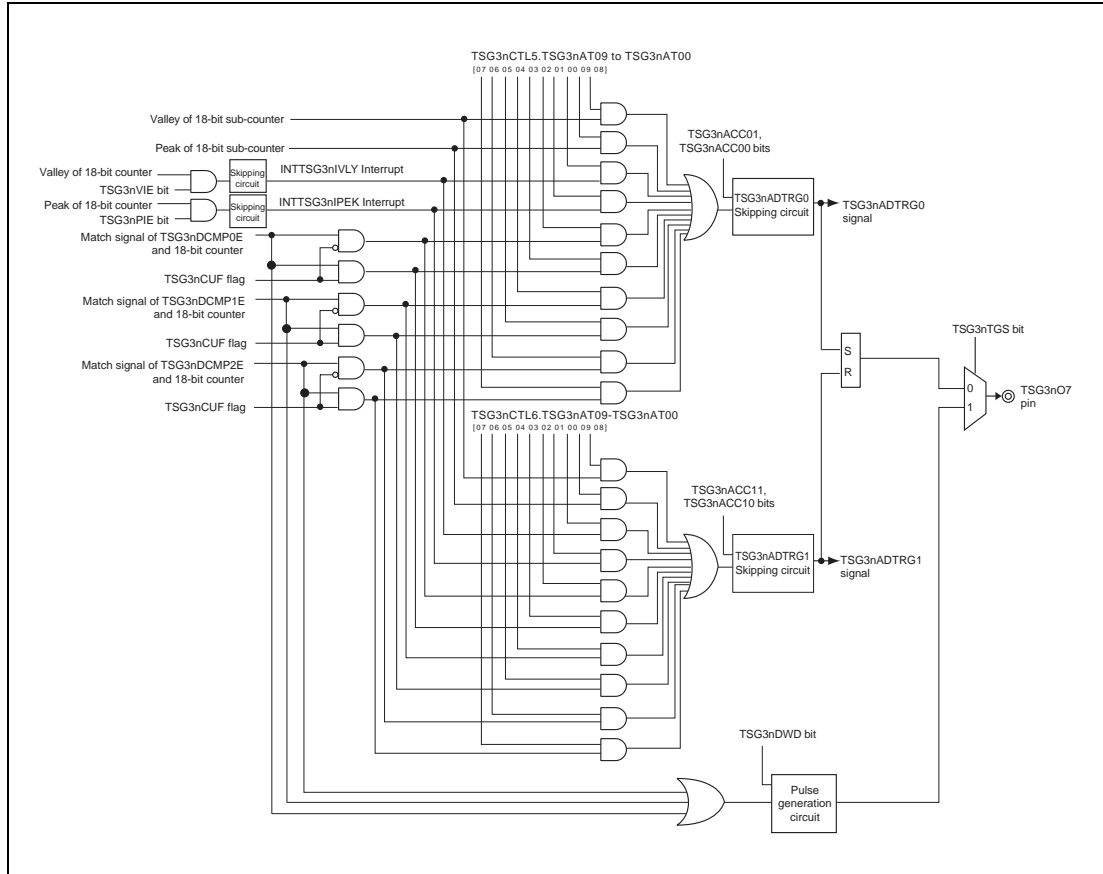


Figure 25.40 A/D Conversion Trigger and Diagnostic Output Control Circuit

As shown in **Figure 25.40**, a logical ORed signal can be generated by selecting the compare match of TSG3nDCMP0E to TSG3nDCMP2E with the 18-bit counter, a peak interrupt (INTTSG3nIPEK) and a valley interrupt (INTTSG3nIVLY) of the 18-bit counter, the peak timing of 18-bit sub-counter, and the valley timing of 18-bit sub-counter.

TSG3n has two channels of the identical A/D conversion trigger control circuits, which can be controlled independently. TSG3n also provides the A/D conversion trigger skipping function with the skipping rate of 1/1 (no skipping), 1/2, 1/4, or 1/8.

25.4.5.1 Operation of A/D Conversion Trigger

TSG3n has a function to generate A/D conversion start triggers (TSG3nADTRG0 and TSG3nADTRG1 signals) by selecting any of ten trigger sources as required. The trigger sources are selected by TSG3nAT09 to TSG3nAT00 in TSG3nCTL5 and TSG3nAT19 to TSG3nAT10 in TSG3nCTL6.

(1) TSG3nADTRG0/TSG3nADTRG1 Signal Output Control (TSG3nCTL5 and TSG3nCTL6)

[Trigger sources]

- TSG3nAT00/TSG3nAT10 = 1 : A valley interrupt (INTTSG3nIVLY) causes an A/D conversion trigger pulse to be generated.
- TSG3nAT01/TSG3nAT11 = 1 : A peak interrupt (INTTSG3nIPEK) causes an A/D conversion trigger pulse to be generated.
- TSG3nAT02/TSG3nAT12 = 1 : While the 18-bit counter is counting up, a TSG3nDCMP0E compare match enables A/D conversion trigger to be generated.
- TSG3nAT03/TSG3nAT13 = 1 : While the 18-bit counter is counting down, a TSG3nDCMP0E compare match enables A/D conversion trigger to be generated.
- TSG3nAT04/TSG3nAT14 = 1 : While the 18-bit counter is counting up, a TSG3nDCMP1E compare match enables A/D conversion trigger to be generated.
- TSG3nAT05/TSG3nAT15 = 1 : While the 18-bit counter is counting down, a TSG3nDCMP1E compare match enables A/D conversion trigger to be generated.
- TSG3nAT06/TSG3nAT16 = 1 : While the 18-bit counter is counting up, a TSG3nDCMP2E compare match enables A/D conversion trigger to be generated.
- TSG3nAT07/TSG3nAT17 = 1 : While the 18-bit counter is counting down, a TSG3nDCMP2E compare match enables A/D conversion trigger to be generated.
- TSG3nAT08/TSG3nAT18 = 1 : A valley timing of the 18-bit sub-counter (at a switch from decrementing to incrementing) enables A/D conversion trigger to be generated.
- TSG3nAT09/TSG3nAT19 = 1 : A peak timing of 18-bit sub-counter (at a switch from incrementing to decrementing) enables A/D conversion trigger to be generated.

[Skipping setting]

- TSG3nACC01, TSG3nACC00 and TSG3nACC11, TSG3nACC10 :
Set the skipping rate of TSG3nADTRG0/TSG3nADTRG1

All A/D conversion triggers selected by TSG3nAT09 to TSG3nAT00 and TSG3nAT19 to TSG3nAT10 are logically ORed, and the resultant signals are subjected to skipping control specified by TSG3nACC01 and TSG3nACC00, and TSG3nACC11 and TSG3nACC10, and then the TSG3nADTRG0 and TSG3nADTRG1 signals are generated.

A peak interrupt (INTTSG3nIPEK) and a valley interrupt (INTTSG3nIVLY) selected by TSG3nAT00 and TSG3nAT01, and TSG3nAT10 and TSG3nAT11 are interrupt signals obtained after skipping. Therefore, they are output at the timing according to interrupt skipping control. If the interrupt output is not enabled by TSG3nCTL4.TSG3nPIE and TSG3nVIE, A/D conversion trigger is not output.

TSG3nACC01, TSG3nACC00, and TSG3nAT09 to TSG3nAT00, and TSG3nACC11, TSG3nACC10, and TSG3nAT19 to TSG3nAT10 can be rewritten during timer operation.

If A/D conversion trigger setting bits are rewritten during operation, the rewritten values are reflected on the A/D conversion trigger output status immediately. Such control bits are rewritten anytime regardless of operating modes. If a write access is made to TSG3nCTL5 and TSG3nCTL6 (including a rewrite of the same value), the A/D conversion trigger skipping counter is cleared and starts counting from zero.

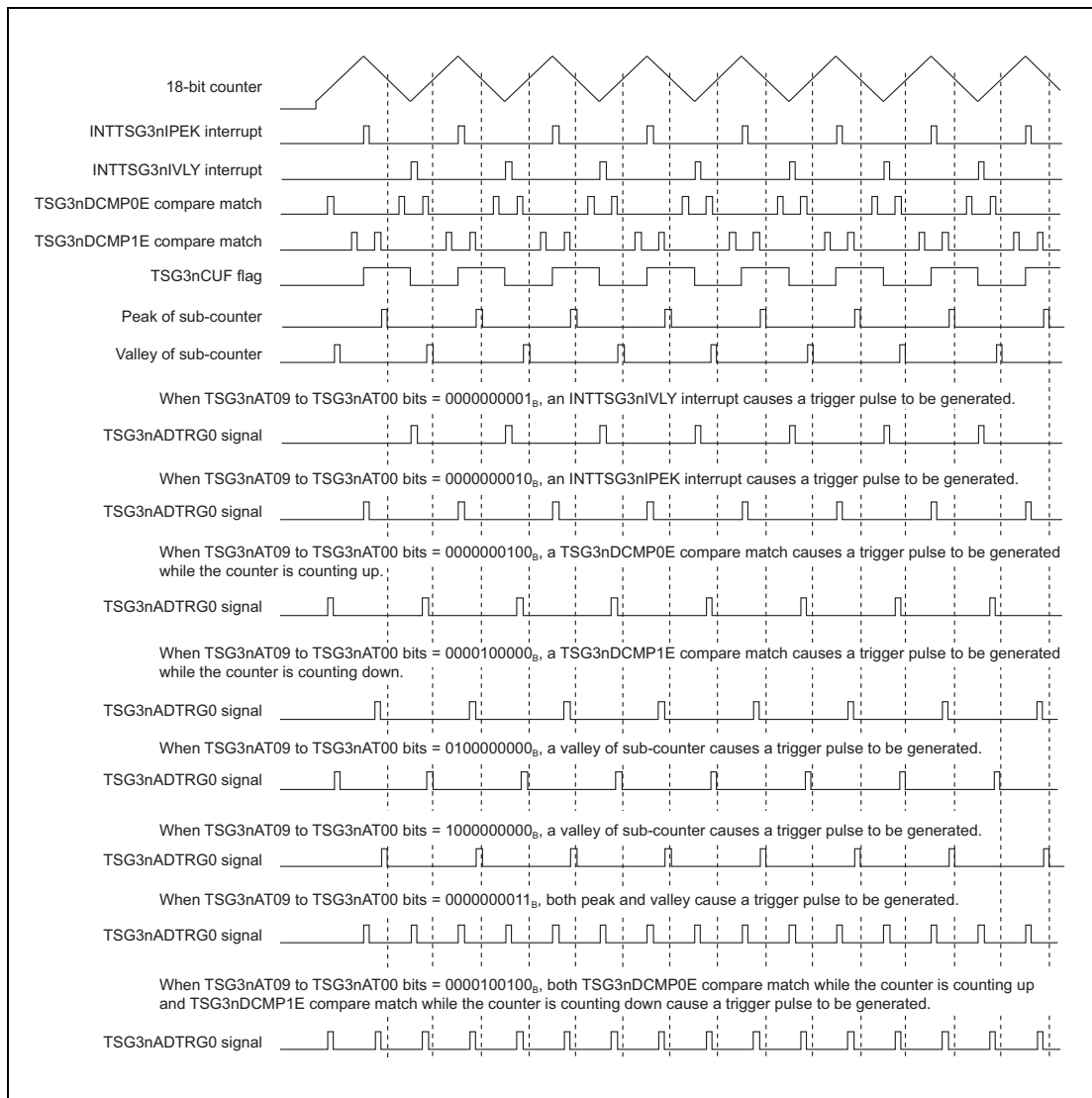


Figure 25.41 When TSG3nPIE = 1, TSG3nVIE = 1, and TSG3nRCC04 to TSG3nRCC00 = 00_H in TSG3nCTL4, and TSG3nACC01 and TSG3nACC00 = 00_B in TSG3nCTL5 (HT-PWM Mode)

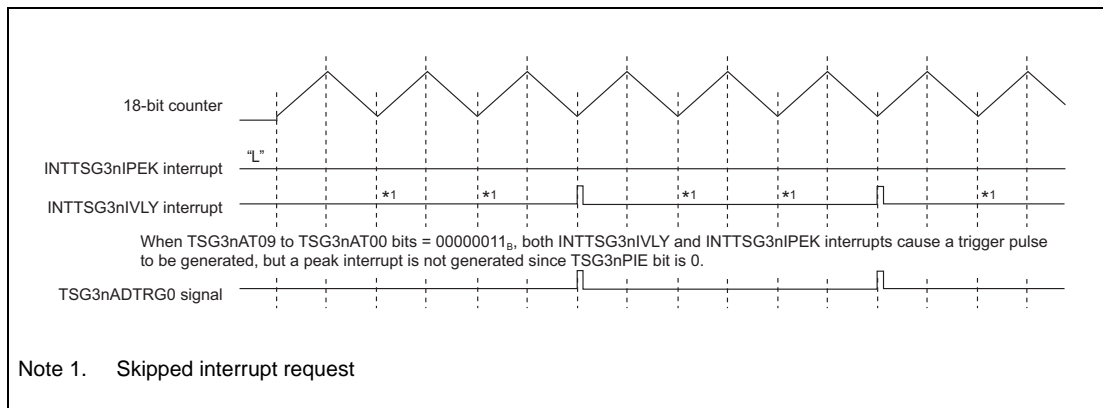


Figure 25.42 When TSG3nPIE = 0, TSG3nVIE = 1, and TSG3nRCC04 to TSG3nRCC00 = 02_H in TSG3nCTL4 and TSG3nACC01 and TSG3nACC00 = 00_B in TSG3nCTL5 (HT-PWM Mode)

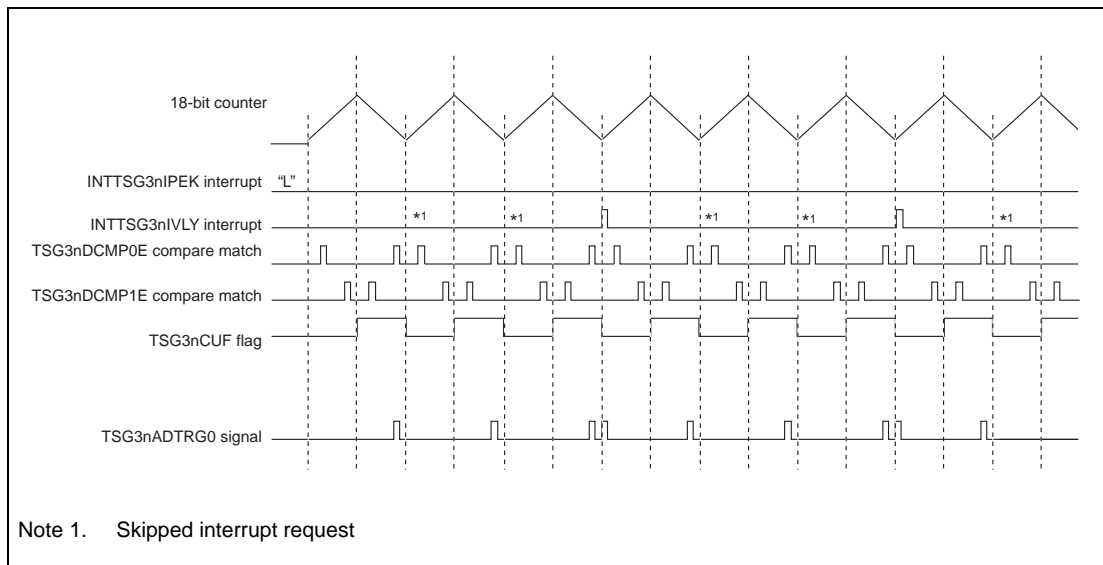


Figure 25.43 When TSG3nPIE = 0, TSG3nVIE = 1, and TSG3nRCC04 to TSG3nRCC00 = 02_H in TSG3nCTL4 and TSG3nACC01 and TSG3nACC00 = 00_B, and TSG3nAT09 to TSG3nAT00 = 0000 1001_B in TSG3nCTL5 (HT-PWM Mode)

(2) A/D Conversion Trigger Skipping Function

Example of operation of the A/D conversion trigger skipping function is shown in **Figure 25.44**.

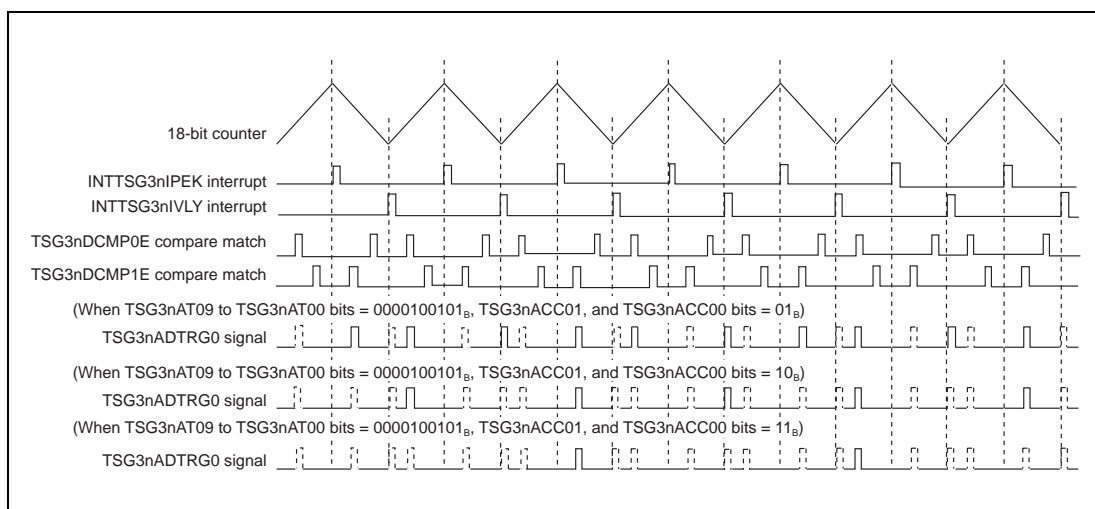


Figure 25.44 Example of Operation of A/D Conversion Trigger Skipping Function

NOTE

Broken-lined pulses indicate A/D conversion trigger pulses skipped by the A/D conversion trigger skipping function.

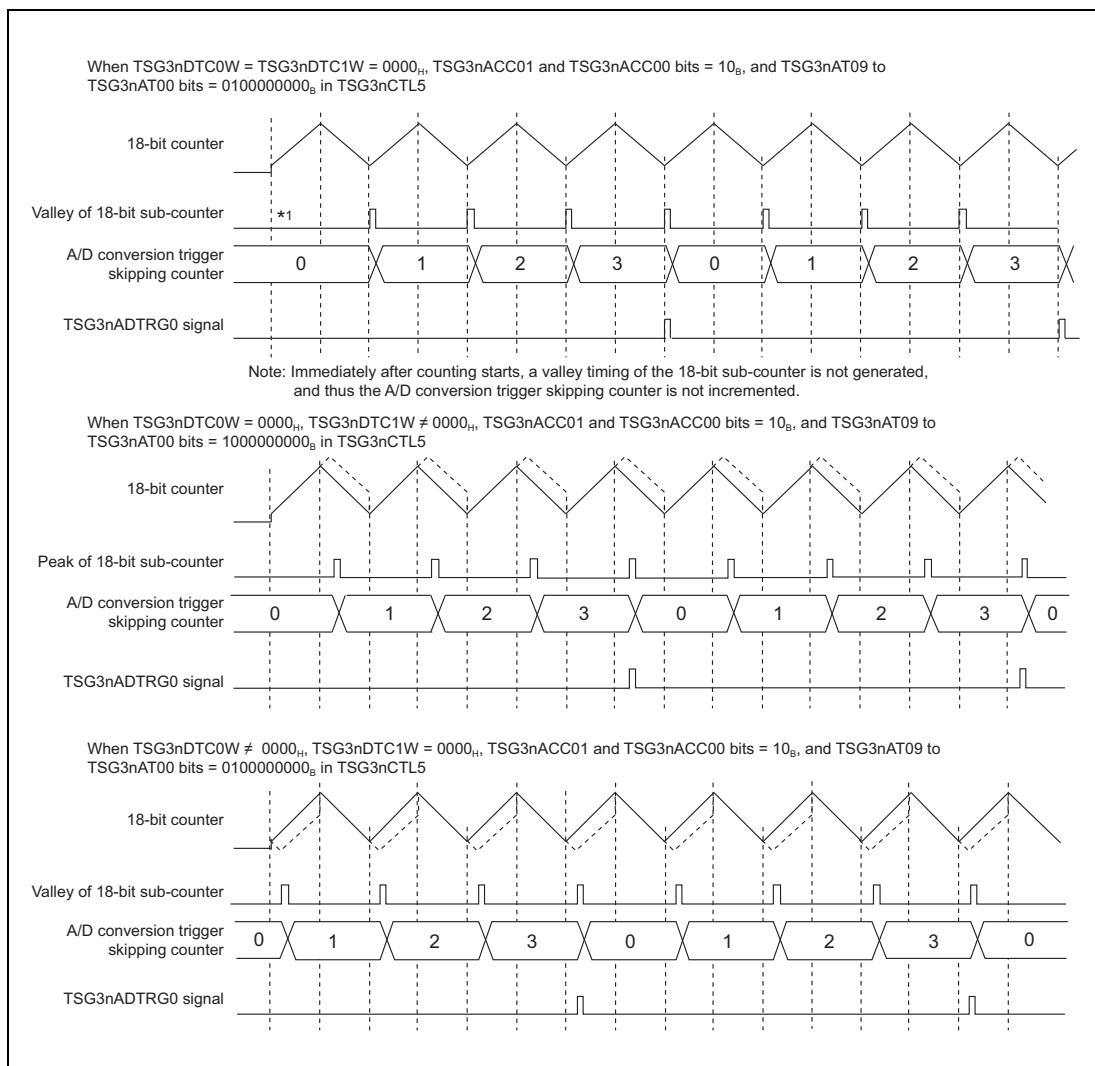


Figure 25.45 Example of Operation of A/D Conversion Trigger Skipping Function

(3) Notes on A/D Conversion Trigger

- If the same value is written to TSG3nDCMP0E and TSG3nDCMP1E or TSG3nDCMP2E, and the same condition (when the 18-bit counter increments or decrements) is set as the valid A/D conversion trigger, A/D conversion trigger skipping counter is incremented by one and one trigger pulse is output upon a match of the 18-bit counter with these registers.
- In PWM mode, SP-PWM mode, 120-DC mode, and HSP-PWM mode, a valley interrupt (INTTSG3nIVLY) is not generated. Only a peak interrupt (INTTSG3nIPEK) is valid.
- In 120-DC mode, when TSG3nS120DCO is set to 0, the 18-bit counter may be cleared during the carrier period due to switch of the output pattern. The A/D conversion trigger is not generated if TSG3nDCMP2E to TSG3nDCMP0E values do not match with the 18-bit counter value and a peak interrupt (INTTSG3nIPEK) is not generated.

25.4.6 Error/Warning Interrupt

25.4.6.1 Error Interrupt Function

If the simultaneous active state of the positive phase and inverse phase is detected after the error interrupt function is enabled ($TSG3nIOC1.TSG3nEOC = 1$), $TSG3nSTR2.TSG3nTBF$ is set, and an error interrupt ($INTTSG3nIER$) of TSG3n is generated. Whether or not to detect an error of each phase ($TSG3nO1$ and $TSG3nO2$, $TSG3nO3$ and $TSG3nO4$, and $TSG3nO5$ and $TSG3nO6$ pins) can be selected by $TSG3nCTL1.TSG3nTBA2$ to $TSG3nTBA0$, respectively.

When an error occurs, outputs of the $TSG3nO1$ to $TSG3nO6$ pins can be set to high-impedance. For details, see **Section 26.4.1, Asynchronous Hi-Z Control Function**.

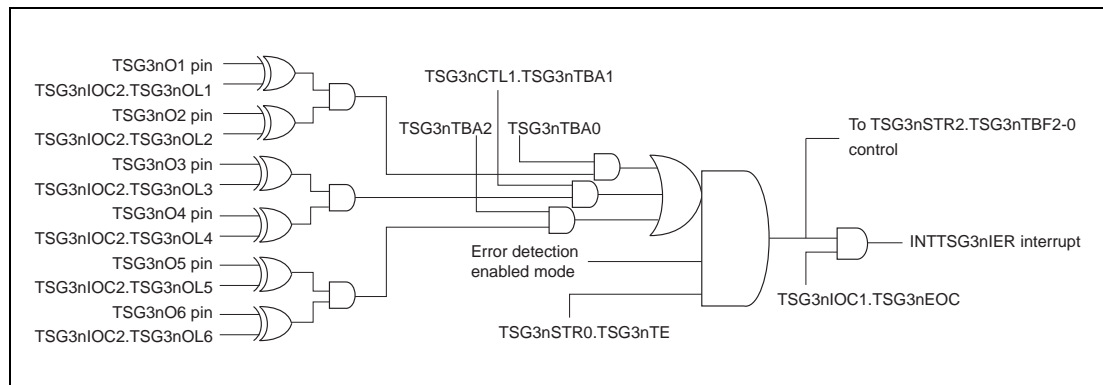


Figure 25.46 Error Interrupt ($INTTSG3nIER$) Generation Control Circuit

CAUTION

When an error interrupt is generated, the error status should be canceled (write 1 to $TSG3nSTC.TSG3nTBR2$ to $TSG3nSTC.TSG3nTBR0$) during error interrupt servicing. Otherwise, subsequent error interrupts are not generated.

(1) PWM Mode, 120-DC Mode and HSP-PWM Mode

In PWM mode and HSP-PWM mode, if TSG3nCMP1E and TSG3nCMP2E, and TSG3nCMP3E and TSG3nCMP4E are set so that the TSG3nO1 and TSG3nO2 pins output the active level simultaneously, an error interrupt (INTTSG3nIER) is generated. Likewise, if TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP9E, TSG3nCMP10E, TSG3nCMP11E, and TSG3nCMP12E are set so that the TSG3nO3 and TSG3nO4, and TSG3nO5 and TSG3nO6 pins output the active level simultaneously, an error interrupt (INTTSG3nIER) is generated.

In 120-DC mode, if TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, TSG3nCMP10E, TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, TSG3nCMP12E, TSG3nPAT0W, and TSG3nPAT1W are set so that the TSG3nO1 and TSG3nO2 pins output the active level simultaneously, an error interrupt (INTTSG3nIER) is generated. With the same setting, the TSG3nO3 and TSG3nO4, and TSG3nO5 and TSG3nO6 pins also output the active level simultaneously and an error interrupt (INTTSG3nIER) is generated.

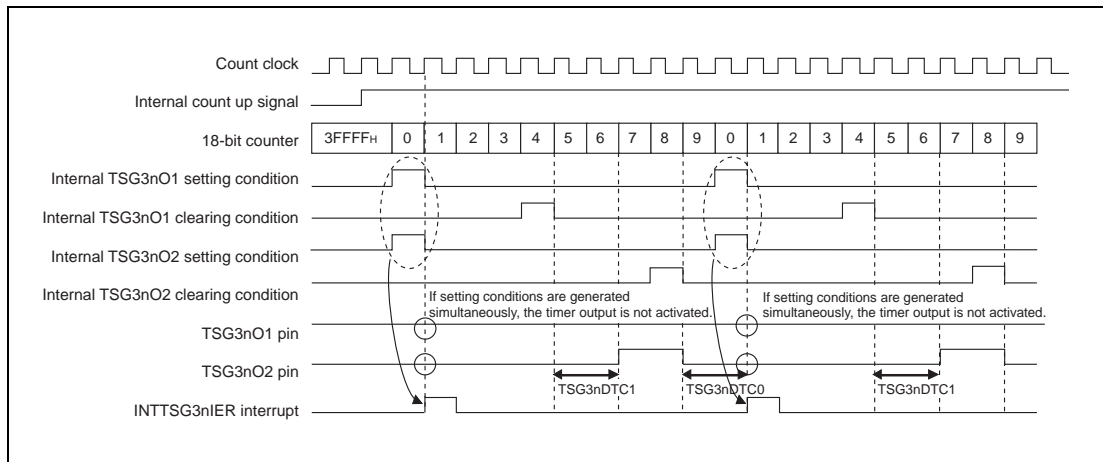


Figure 25.47 Example of Error Interrupt (INTTSG3nIER) Generation (PWM Mode)

NOTE

TSG3nO3 and TSG3nO4, and TSG3nO5 and TSG3nO6 behave the same.

When the active level of output is switched by manipulating TSG3nIOC2.TSG3nOL1 and TSG3nOL2, an error interrupt is generated as shown in the following figure.

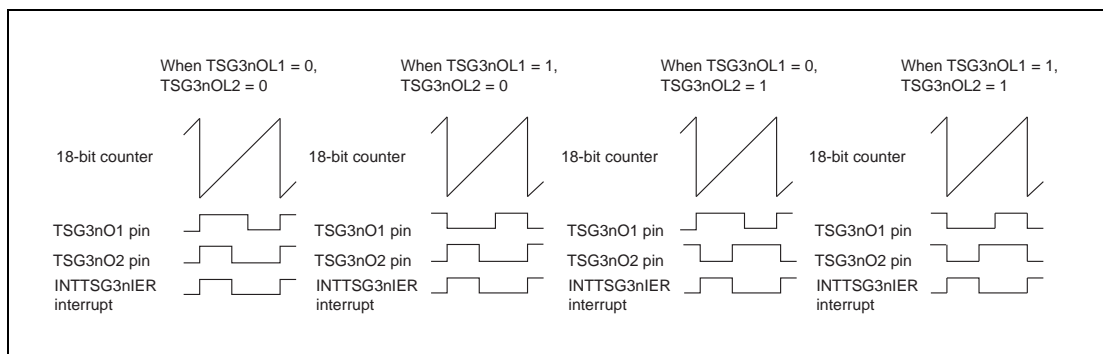


Figure 25.48 Example of Error Interrupt (INTTSG3nIER) Generation for each Active Level

(2) HT-PWM Mode and SP-PWM Mode

When either TSG3n dead time control register 0 or 1 (TSG3nDTC0W or TSG3nDTC1W) is 0000_H, an error may occur.

NOTE

If an error occurs when the dead time control function is used (both TSG3nDTC0W and TSG3nDTC1W are not 0000_H), internal circuit failure may have occurred.

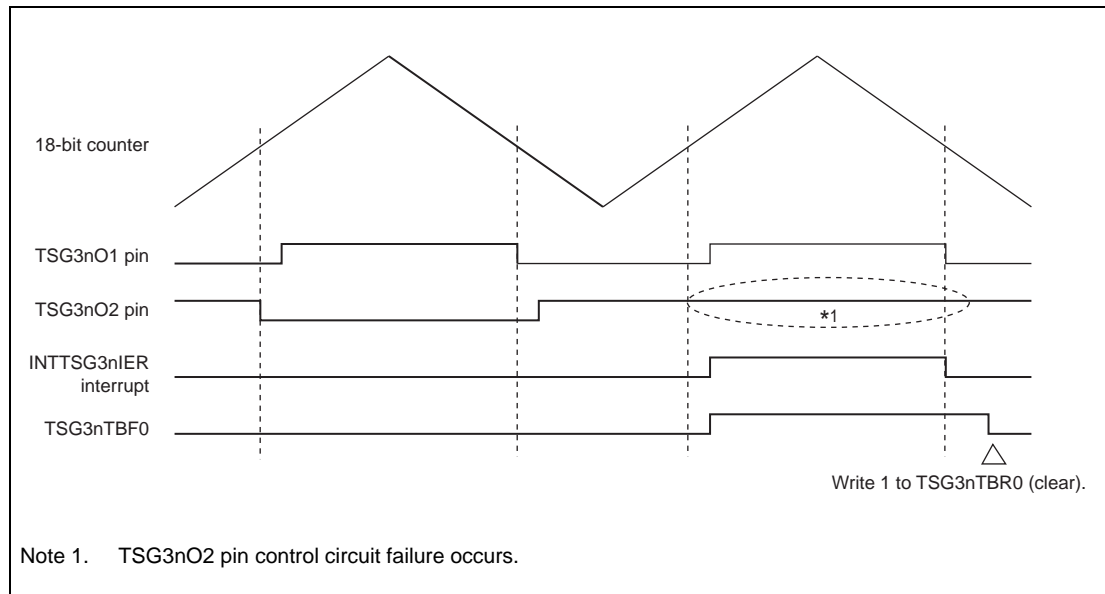


Figure 25.49 Example of Error Interrupt Operation

25.4.6.2 Warning Interrupt Function

TSG3n has a warning interrupt (INTTSG3nIWN).

Warning interrupt (INTTSG3nIWN) is generated when any of the following conditions is detected.

For details, see **Section 25.4.3, Flags**.

- When simultaneous change in two or more pins of TSG3nPTSI2 to TSG3nPTSI0 is detected:
See **Section 25.4.3.4, Noise Detection Flag (TSG3nNDF)**.
- When reversal is detected of the TSG3nPTSI2 to TSG3nPTSI0 pins:
See **Section 25.4.3.7, Pattern Reversal Detection Flag (TSG3nPRF)**.
- When 000 or 111 is detected from the TSG3nPTSI2 to TSG3nPTSI0 pins:
See **Section 25.4.3.6, Pattern Error Detection Flag (TSG3nPEF)**.
- When a toggle of the TSG3nPTSI2 to TSG3nPTSI0 pins is generated three or more times between TSG3nOPCI0 and TSG3nOPCI1 signal triggers.:
See **Section 25.4.3.8, TSG3nPTSI2 to TSG3nPTSI0 Pin Abnormal Toggle Detection Flag (TSG3nPTF)**.
- When the TSG3nOPCI0 and TSG3nOPCI1 signal triggers are detected simultaneously:
See **Section 25.4.3.9, TSG3nOPCI0 and TSG3nOPCI1 Signal Simultaneous Trigger Detection Flag (TSG3nTDF)**.
- When the phase difference between the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) and output pattern (TSG3nOPF2 to TSG3nOPF0) is detected:
See **Section 25.4.3.10, Pattern Phase Difference Detection Flag (TSG3nPPF)**.

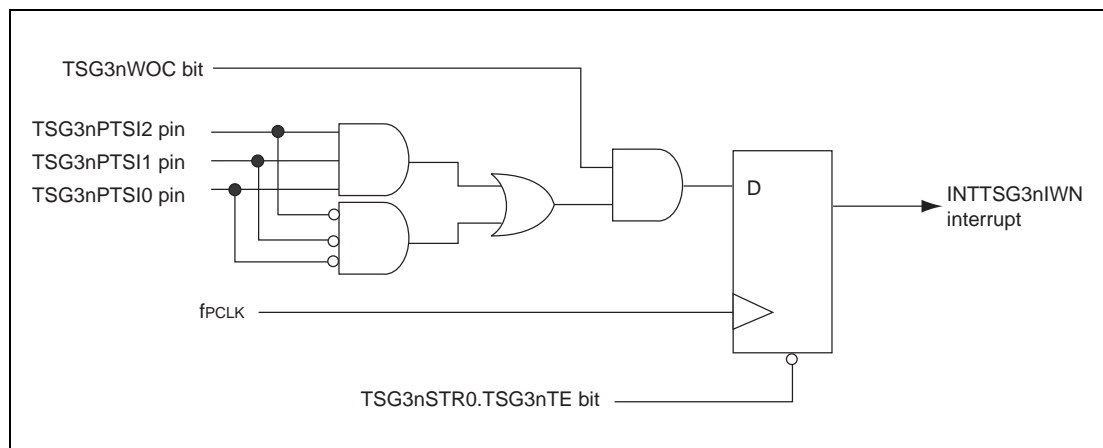


Figure 25.50 Detection of Abnormality of TSG3nPTSI2-TSG3nPTSI0 Pins

25.4.7 Operating Modes

25.4.7.1 PWM Mode

Overview

A PWM signal is output at the TSG3nO1 to TSG3nO6 pins according to set timing/clear timing of TSG3nCMP1E to TSG3nCMP12E registers with the PWM period set in the TSG3nCMP0E register.

Prerequisites

- Set the set timing to the compare register with an even number:
TSG3nCMP2E (set timing of the TSG3nO1 output), TSG3nCMP4E (set timing of the TSG3nO2 output), TSG3nCMP6E (set timing of the TSG3nO3 output), TSG3nCMP8E (set timing of the TSG3nO4 output), TSG3nCMP10E (set timing of the TSG3nO5 output) and TSG3nCMP12E (set timing of the TSG3nO6 output)
- Set the clear timing to the compare register with an odd number:
TSG3nCMP1E (clear timing of the TSG3nO1 output), TSG3nCMP3E (clear timing of the TSG3nO2 output), TSG3nCMP5E (clear timing of the TSG3nO3 output), TSG3nCMP7E (clear timing of the TSG3nO4 output), TSG3nCMP9E (clear timing of the TSG3nO5 output) and TSG3nCMP11E (clear timing of the TSG3nO6 output)

Functional description

Set the PWM period and set/clear timing of the TSG3nO1 to TSG3nO6 outputs. Set TSG3nTRG0.TSG3nTS = 1 to start the timer counter.

The TSG3nO1 to TSG3nO6 outputs are set to the inactive state at the same time the counting begins. The outputs are set to the active state by the match of the buffer registers TSG3nCMP2E, TSG3nCMP4E, TSG3nCMP6E, TSG3nCMP8E, TSG3nCMP10E, and TSG3nCMP12E with the 18-bit counter.

Next, the TSG3nO1 to TSG3nO6 outputs are set to the inactive state by the match of the buffer registers TSG3nCMP1E, TSG3nCMP3E, TSG3nCMP5E, TSG3nCMP7E, TSG3nCMP9E, and TSG3nCMP11E with the 18-bit counter.

During counting, a compare match interrupt (INTTSG3nI0 to INTTSG3nI12) is generated by the match of the buffer register TSG3nCMP0E-TSG3nCMP12E with the 18-bit counter.

CAUTION

Reload is executed when a value is written to the TSG3nCMP1E register while TSG3nCTL3.TSG3nRMC = 0. Therefore, even when it is needed to rewrite only the value of the TSG3nCMP0E register, a write operation to the TSG3nCMP1E register is necessary. When only the TSG3nCMP0E register is rewritten, reload is not done.

NOTE

The PWM mode is set when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 000_B.

(a) When TSG3nCMP0E and TSG3nCMP1E to TSG3nCMP12E are Not Rewritten during Timer Operation

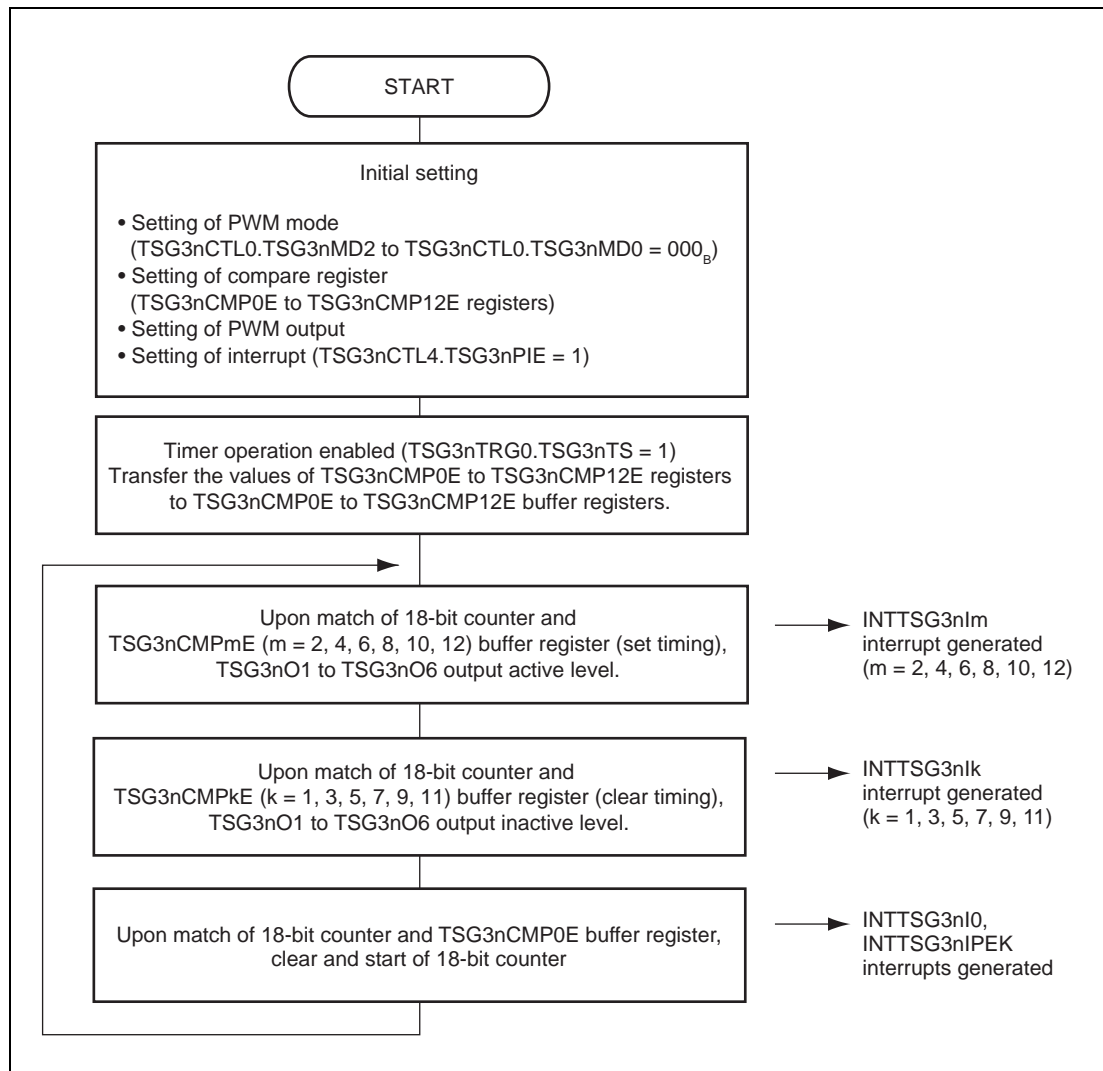


Figure 25.51 Basic Operation Flow of PWM Mode (1/2)

(b) When TSG3nCMP0E and TSG3nCMP1E to TSG3nCMP12E are Rewritten during Timer Operation

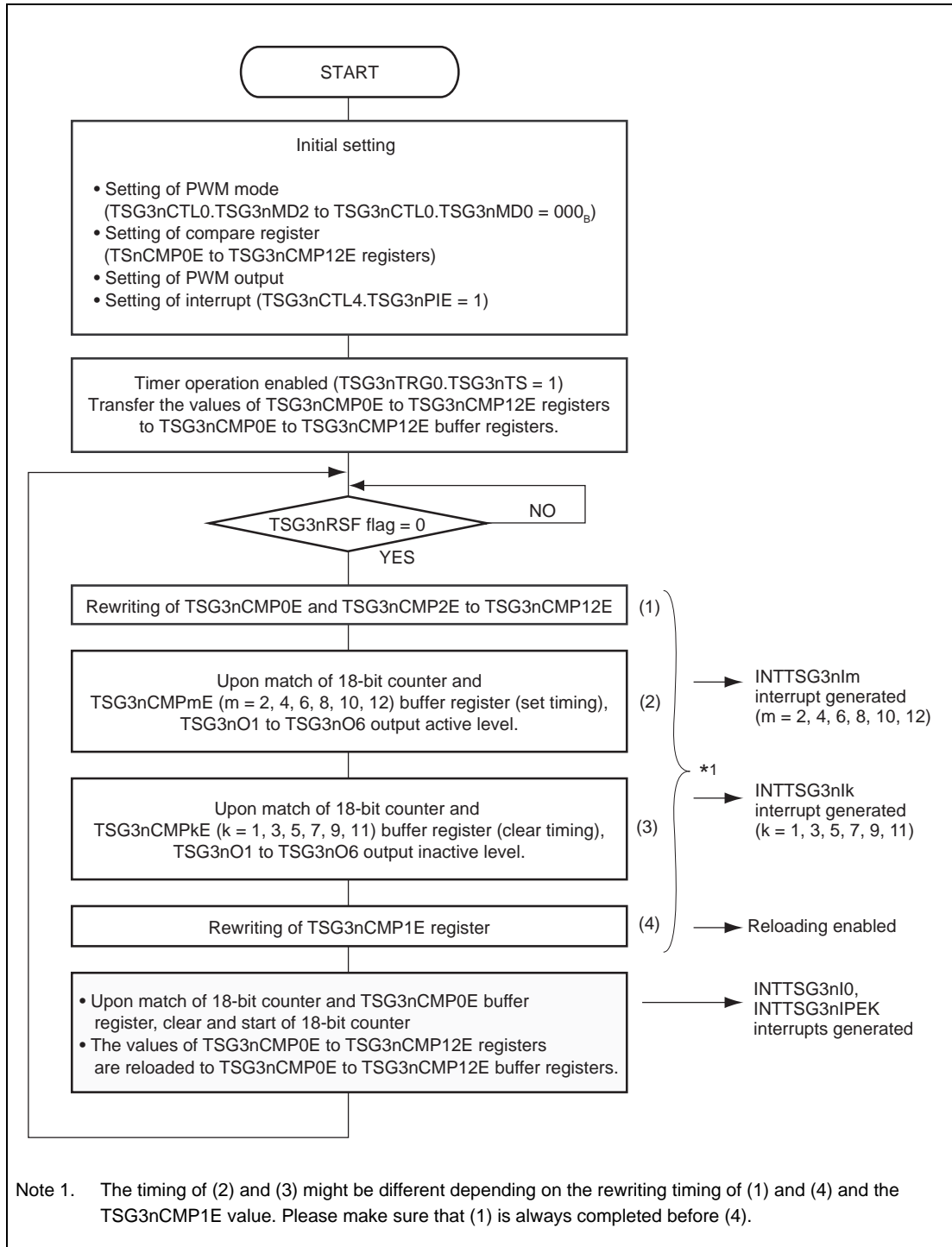


Figure 25.51 Basic Operation Flow of PWM Mode (2/2)

CAUTION

Please rewrite compare registers after confirming that the reload request flag TSG3nRSF is 0.

(1) List of Operations in PWM Mode**Table 25.53 Counter Functions in PWM Mode**

| Operation | | Setting condition |
|----------------|-------|--|
| 18-bit counter | Start | TSG3nTRG0.TSG3nTS = 0 → 1, or a simultaneous start trigger |
| | Clear | Compare match of TSG3nCMP0E buffer register and 18-bit counter |
| | Stop | TSG3nTRG1.TSG3nTT = 0 → 1 |

Table 25.54 Functions of Compare Registers and Dead Time Control Register in PWM Mode

| Register | Rewriting Method | Rewrite during Operation | Function |
|---|------------------------|--------------------------|---|
| TSG3nCMP0E | Reload/Anytime rewrite | Possible | Setting period |
| TSG3nCMPmE (m = 1 to 12) | Reload/Anytime rewrite | Possible | Setting set/clear timing |
| TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E | Reload/Anytime rewrite | Possible | Diagnostic output or A/D conversion trigger |
| TSG3nDTC0W, TSG3nDTC1W | Reload | Possible* ¹ | Setting dead time |

Note 1. For the detail see, **(3), Controlling Dead Time in PWM Mode.**

Table 25.55 Timer Input Function in PWM Mode

| Pin | Function |
|-----------|--------------------|
| TSG3nCLKI | Clock enable input |

Table 25.56 Timer Output Function in PWM Mode

| Pin | Function |
|----------------------|--|
| TSG3nOm (m = 1 to 6) | PWM output by compare match of TSG3nCMPkE buffer register and 18-bit counter (k = 1 to 12) |
| TSG3nO7 | Diagnostic signal output or pulse output by A/D conversion trigger |

Table 25.57 Interrupt Requests in PWM Mode

| Interrupt | Function |
|--------------------------|---|
| INTTSG3nIm (m = 0 to 12) | Compare match of TSG3nCMPmE buffer register and 18-bit counter (m = 0 to 12) |
| INTTSG3nIER | Error (detection of simultaneous active state of TSG3nO1 and TSG3nO2, or TSG3nO3 and TSG3nO4, or TSG3nO5 and TSG3nO6) |
| INTTSG3nIVLY | — |
| INTTSG3nIPEK | Peak interrupt (generated at the same timing as INTTSG3nI0) |
| INTTSG3nIWN | Warning interrupt |

Note: “—”: Not available in PWM mode

Table 25.58 Compare Match Timing in PWM Mode

| Compare Match | Timing |
|--------------------------|--|
| TSG3nCMP0E | When 18-bit counter changes from TSG3nCMP0E to 00000 _H |
| TSG3nCMPmE (m = 1 to 12) | After detecting the match of 18-bit counter and TSG3nCMPmE (m = 1 to 12) |

Table 25.59 Example of Setting each Timer Output Condition in PWM Mode

| Pin | Item | Output Period | Output Duty | |
|-------------------------|---------------|-----------------------------------|--|---|
| | | | Output Condition | Setting Condition |
| TSG3nOm (m = 1 to 6) | PWM output | (TSG3nCMP0E + 1) x count clock | Output an inactive level throughout one period (duty cycle 0%) | TSG3nCMPmE = TSG3nCMP (m + 1)E or TSG3nCMP (m + 1)E > TSG3nCMP0E (m = 1, 3, 5, 7, 9, 11) |
| | | | Output an active level of one count clock in one period | TSG3nCMPmE = TSG3nCMP (m + 1)E + 1 TSG3nCMP (m + 1)E = TSG3nCMPmE - 1 (m = 1, 3, 5, 7, 9, 11) |
| | | | Output an inactive level of one count clock in one period | TSG3nCMPmE = TSG3nCMP (m + 1)E - 1 TSG3nCMP (m + 1)E = TSG3nCMPmE + 1 (m = 1, 3, 5, 7, 9, 11) |
| | | | Output an active level throughout one period (duty cycle 100%) | TSG3nCMPmE > TSG3nCMP0E TSG3nCMP (m + 1)E ≤ TSG3nCMP0E (m = 1, 3, 5, 7, 9, 11) |

**When only TSG3nCMP2E is rewritten and TSG3nO1 is output
(TSG3nIOC0.TSG3nTOE1 = 1, TSG3nIOC2.TSG3nOL1 = 0)**

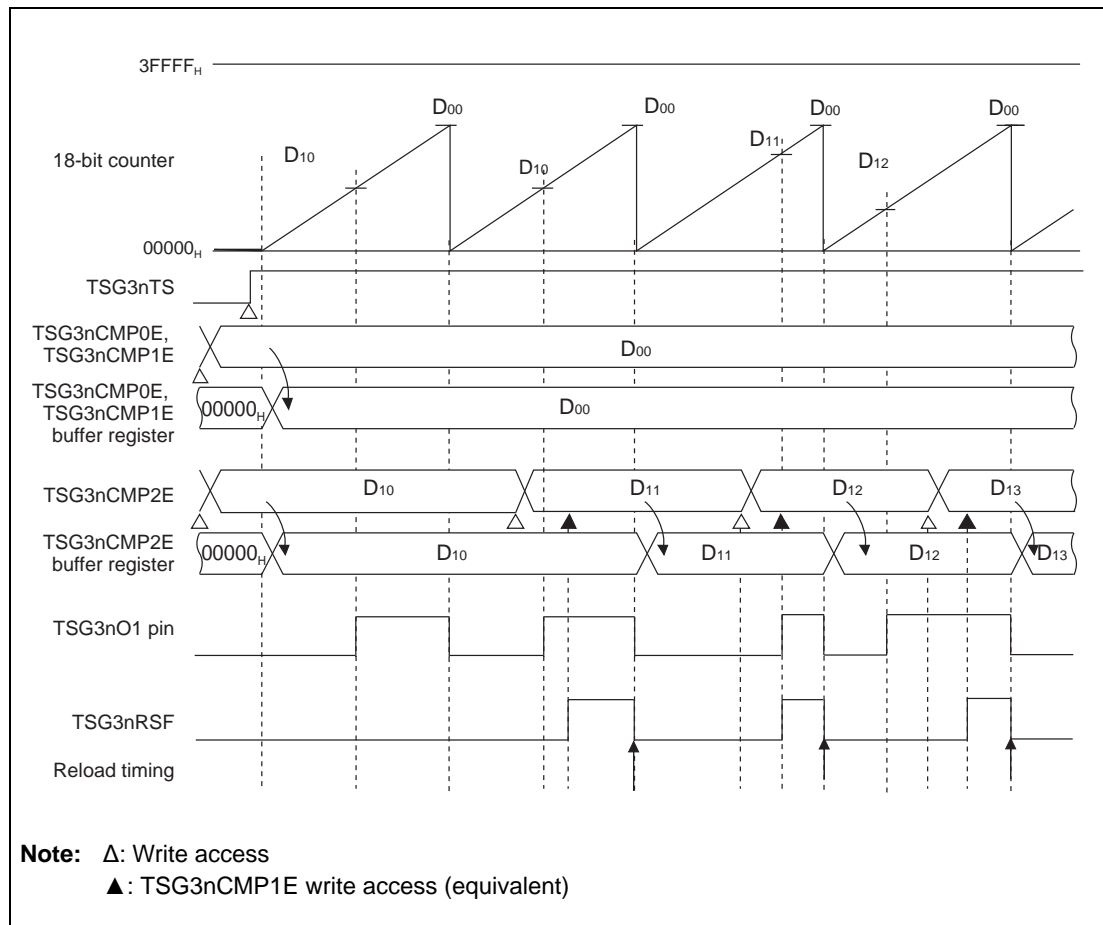


Figure 25.52 Example of Basic Operation Timing of PWM Mode (1/2)

NOTES

1. D00: Set values of TSG3nCMP0E and TSG3nCMP1E (00000_H-3FFFF_H)
D10, D11, D12 and D13: Set values of TSG3nCMP2E (00000_H-3FFFF_H)
2. TSG3nO1 (PWM) duty cycle = (TSG3nCMP1E-TSG3nCMP2E)×(count clock cycle)
TSG3nO1 (PWM) period = (Set value of TSG3nCMP0E + 1)×(count clock cycle)
3. TSG3nO2-TSG3nO6 pins behave similarly to the TSG3nO1 pin

When TSG3nCMP0E-TSG3nCMP2E are rewritten, and TSG3nO1 is output
(TSG3nIOC0.TSG3nTOE1 = 1, TSG3nIOC2.TSG3nOL1 = 0)

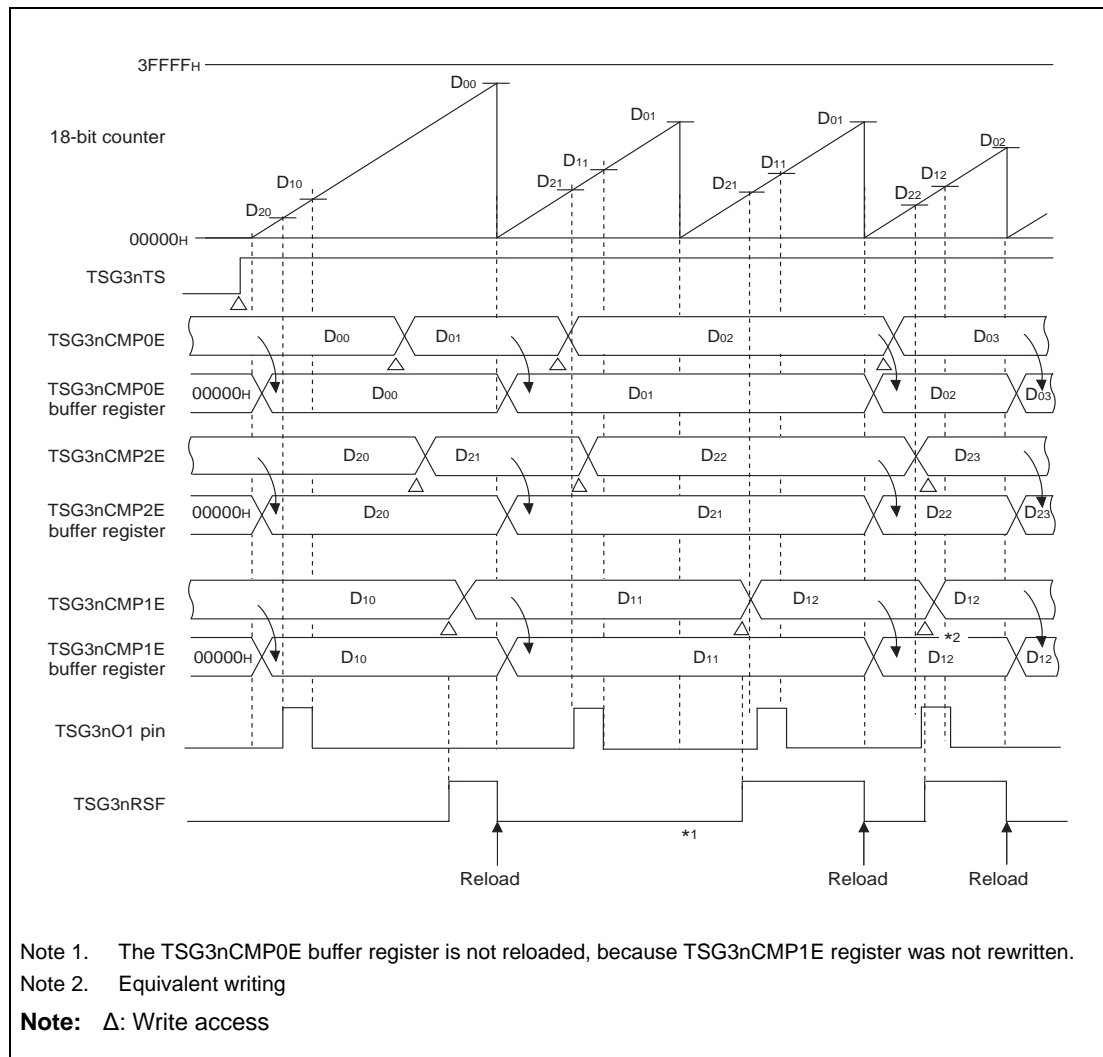


Figure 25.52 Example of Basic Operation Timing of PWM Mode (2/2)

NOTES

- D00, D01, D02, D03: Set value of TSG3nCMP0E (00000_H-3FFFF_H)
D10, D11, D12, D13: Set value of TSG3nCMP1E (00000_H-3FFFF_H)
D20, D21, D22, D23: Set value of TSG3nCMP2E (00000_H-3FFFF_H)
- Outputs from TSG3nO2 to TSG3nO6 behave similarly to the TSG3nO1 pin.

(2) Interrupt/Reload Skipping Function in PWM Mode

By setting TSG3nCTL4.TSG3nPRE and TSG3nPIE to 1 and setting the TSG3nRCC04 to TSG3nRCC00 and TSG3nCTL3.TSG3nRIA, the reload and interrupt skipping function can be used.

By setting TSG3nPRE to 1 and setting the TSG3nRCC04 to TSG3nRCC00, the interrupt skipping function can be used.

(3) Controlling Dead Time in PWM Mode

By setting the dead time values in the TSG3nDTC0W and TSG3nDTC1W registers in PWM mode, it is possible to control the dead time. The dead time is controlled according to the switch timing of the TSG3nO1 and TSG3nO2 outputs, the TSG3nO3 and TSG3nO4 outputs, or the TSG3nO5 and TSG3nO6 outputs.

Table 25.60 Dead Time in PWM Mode

| Switch Timing | Dead Time |
|--|------------------------------|
| TSG3nO1: High level to low level and TSG3nO2: Low level to high level | Value of TSG3nDTC1W register |
| TSG3nO2: High level to low level and TSG3nO1: Low level to high level | Value of TSG3nDTC0W register |
| TSG3nO3: High level to low level and TSG3nO4: Low level to high level | Value of TSG3nDTC1W register |
| TSG3nO4: High level to low level and TSG3nO3: Low level to high level | Value of TSG3nDTC0W register |
| TSG3nO5: High level to low level and TSG3nO6: Low level to high level | Value of TSG3nDTC1W register |
| TSG3nO6: High level to low level and TSG3nO5: Low level to high level | Value of TSG3nDTC0W register |

NOTE

Dead time counter keeps operating even when operation stop (TSG3nTE = 0) setting collided with dead time insert timing, and the dead time set in TSG3nO1 and TSG3nO2 are always inserted.

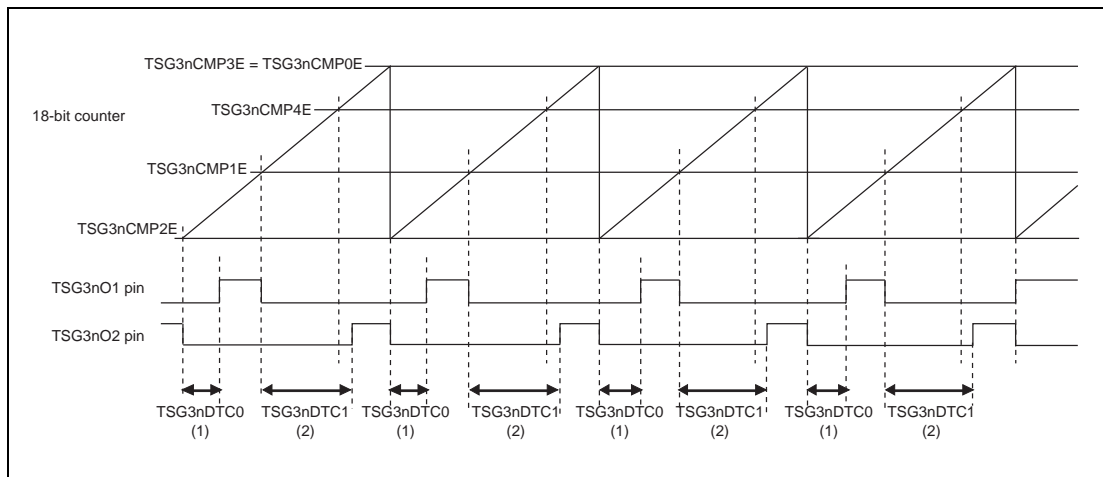


Figure 25.53 Example of Dead Time Control between TSG3nO1 and TSG3nO2 Outputs (1/2)

During (1), the dead time counter starts counting at the falling edge of the TSG3nO2 output. At this time, even after the 18-bit counter reaches 00000_H indicating that the TSG3nO1 output should be active, the TSG3nO1 output stays inactive because the dead time counter is still operating. The TSG3nO1 output becomes active at the timing when the dead time count operation ends.

At (2), the dead time counter starts counting at the falling edge of the TSG3nO1 output. Even after the match of the 18-bit counter and TSG3nCMP4E indicating that the TSG3nO2 output should be active, the TSG3nO2 output stays inactive because the dead time counter is still operating. The TSG3nO2 output becomes active at the timing when the dead time count operation ends.

NOTES

1. The TSG3nO1 and TSG3nO2 pins are set to active high
2. The TSG3nO3 and TSG3nO4 pins and the TSG3nO5 and TSG3nO6 pins outputs behave similarly.

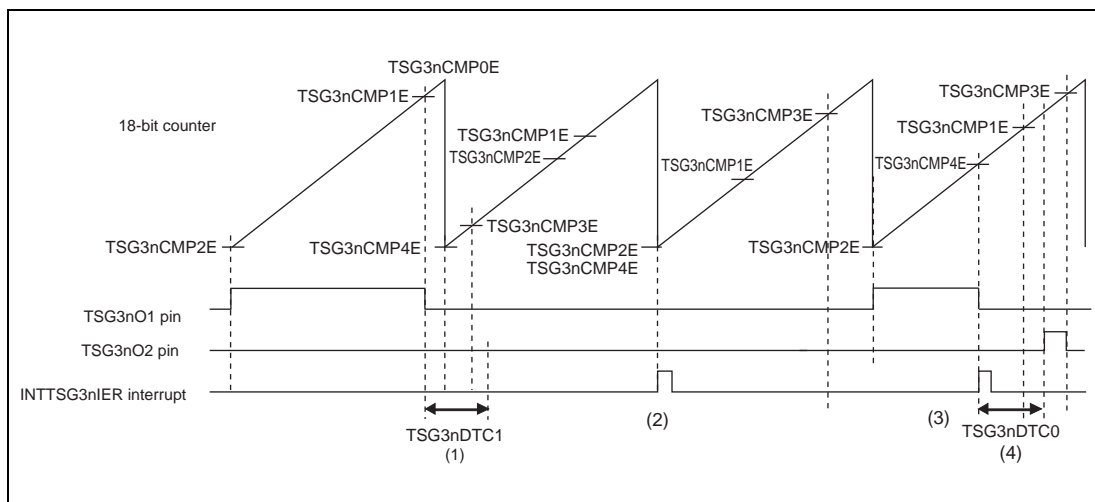


Figure 25.53 Example of Dead Time Control between TSG3nO1 and TSG3nO2 Outputs (2/2)

During (1), the dead time counter starts counting at the falling edge of the TSG3nO1 output. Even after the 18-bit counter reaches 00000_H and the match occurs between the 18-bit counter and TSG3nCMP4E indicating that the TSG3nO2 output should be active, the TSG3nO2 output stays inactive because the dead time counter is still operating. Moreover, since the TSG3nCMP3E register compare match occurs before the operation of the dead time counter ends, the TSG3nO2 output stays inactive.

$$\text{TSG3nCMP1E} + \text{TSG3nDTC1} \geq \text{TSG3nCMP0E} + \text{TSG3nCMP2E}$$

(TSG3nO2 stays inactive)

$$\text{TSG3nCMP2E} + \text{TSG3nDTC0} \geq \text{TSG3nCMP0E} + \text{TSG3nCMP1E}$$

(TSG3nO1 stays inactive)

At (2), the INTTSG3nIER interrupt occurs because the TSG3nCMP2E register and the TSG3nCMP4E register are set so that the TSG3nO1 and TSG3nO2 outputs rise simultaneously. Here, both the TSG3nO1 and TSG3nO2 outputs become inactive.

At (3), compare match with the TSG3nCMP4E register while the TSG3nO1 output is active generates an INTTSG3nIER interrupt and both TSG3nO1 and TSG3nO2 outputs become inactive.

At (4), the falling edge (inactive) of the TSG3nO1 output is caused by the detection of simultaneous active state and the dead time counter starts counting. After the end of the dead time counter operation, the TSG3nO2 output becomes active.

NOTES

1. The TSG3nO1 and TSG3nO2 pins are set to active high.
2. The TSG3nO3 and TSG3nO4 pins and the TSG3nO5 and TSG3nO6 pins outputs behave similarly.

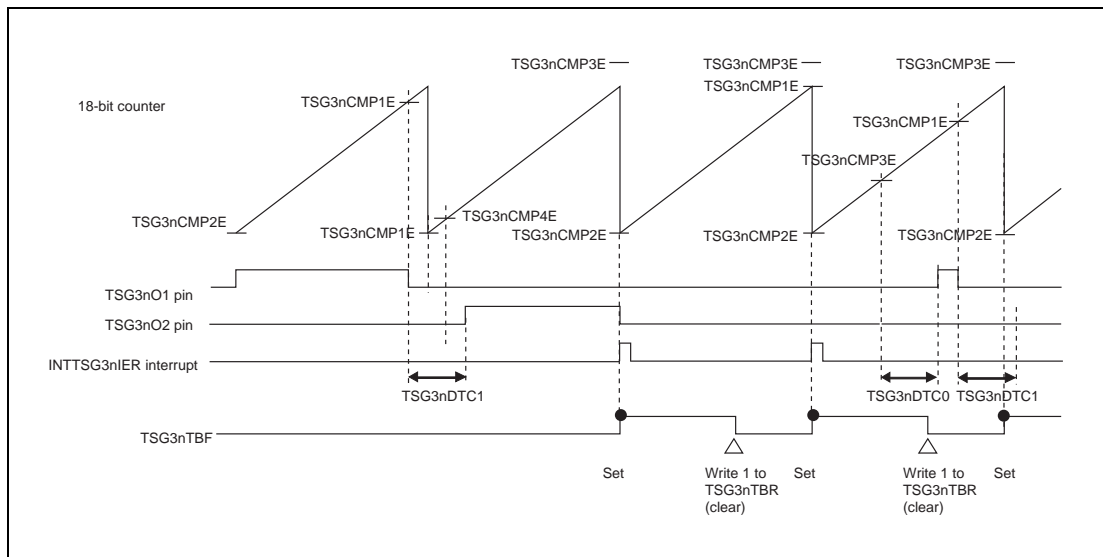


Figure 25.54 Example of 100% Duty Output at Dead Time Control

When the TSG3nO2 pin is set to duty cycle of 100% ($TSG3nCMP3E \geq TSG3nCMP0E + 1$), the output of the TSG3nO1 pin is fixed to a low level. This control is intended to mask the active condition of TSG3nO1 output since the TSG3nO2 output is active before the TSG3nO1 output becomes active. In this case, the INTTSG3nIER interrupt is also generated because TSG3nO1 and TSG3nO2 outputs become high simultaneously.

NOTES

1. The TSG3nO1 and TSG3nO2 pins are set to active high.
2. The TSG3nO3 and TSG3nO4 pins and the TSG3nO5 and TSG3nO6 pins outputs behave similarly.

(4) Dead Time Rewriting during Timer Operation in PWM Mode

In PWM mode, it is possible to rewrite TSG3n dead time control registers TSG3nDTC0W and TSG3nDTC1W while counting. The new settings are active at reload timing. It is not possible to change the dead time setting by rewriting at any time.

To enable reloading, write to the TSG3nCMP1E register.

25.4.7.2 HT-PWM mode (High accuracy Triangular - Pulse Width Modulation mode)

Overview

In this mode, the 18-bit counter (up/down count by ± 2 bits, practically 17 bits) and the 18-bit compare registers (LSB is used to control additional pulse) are used to generate a 6-phase PWM signal.

Prerequisites

- Set the carrier wave period with TSG3nCMP0E.
- Set the duty cycle of the voltage data signals of the U phase, V phase, and W phase with TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE (The values set in TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE are reflected immediately to the corresponding TSG3nCMPmE ($m = 1, 2, 5, 6, 9, 10$)).
- Symmetric triangular wave control is described in this section. Please refer to **Section 25.4.7.2 (10), Asymmetric Triangular Wave Control in HT-PWM Mode**, for asymmetric triangular wave control.

Functional description

In this mode, the carrier period and the duty cycle of the U phase, the V phase, and the W phase are configured. Counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter counts up from TSG3nDTC0 as the minimum value, and counts down upon the match with the maximum value of TSG3nCMP0E + TSG3nDTC0.

The dead time is set with TSG3nDTC0 and TSG3nDTC1. TSG3nDTC0 sets the dead time of inverse phase (OFF) to positive phase (ON) switch and TSG3nDTC1 sets the dead time of positive phase (OFF) to inverse phase (ON) switch. The 10-bit counters (TSG3nDTT1 to TSG3nDTT3) for dead time generation load the set values of TSG3nDTC0 and TSG3nDTC1 by the compare match of the 18-bit counter and the TSG3nCMPm buffer register ($m = 1, 2, 5, 6, 9, 10$), and start down-counting.

INTTSG3nIm interrupts ($m = 1, 2, 5, 6, 9, 10$) are generated by the compare match of the 18-bit counter with TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E buffer registers.

INTTSG3nIm interrupts ($m = 3, 4, 7, 8, 11, 12$) are generated by the compare match of the 18-bit counter with TSG3nCMP3E, TSG3nCMP7E, and TSG3nCMP11E buffer registers when counting down (TSG3nCUF = 1), and with TSG3nCMP4E, TSG3nCMP8E, and TSG3nCMP12E buffer registers when counting up (TSG3nCUF = 0).

NOTE

The HT-PWM mode is enabled when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 001_B.

(1) Block Diagram and Basic Timing Chart

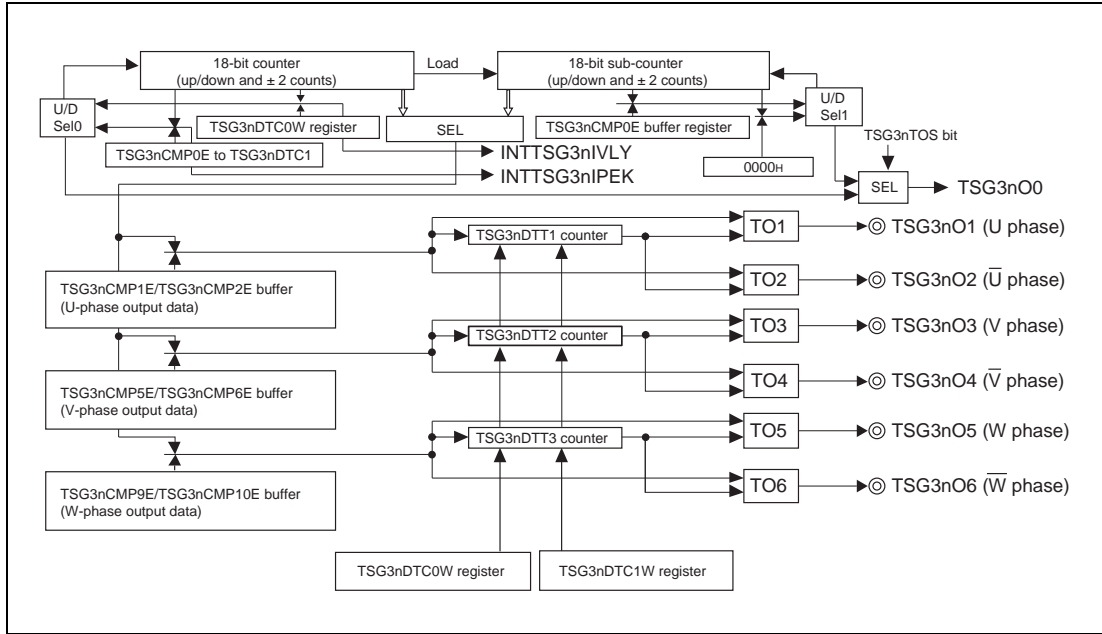


Figure 25.55 Block Diagram in HT-PWM Mode

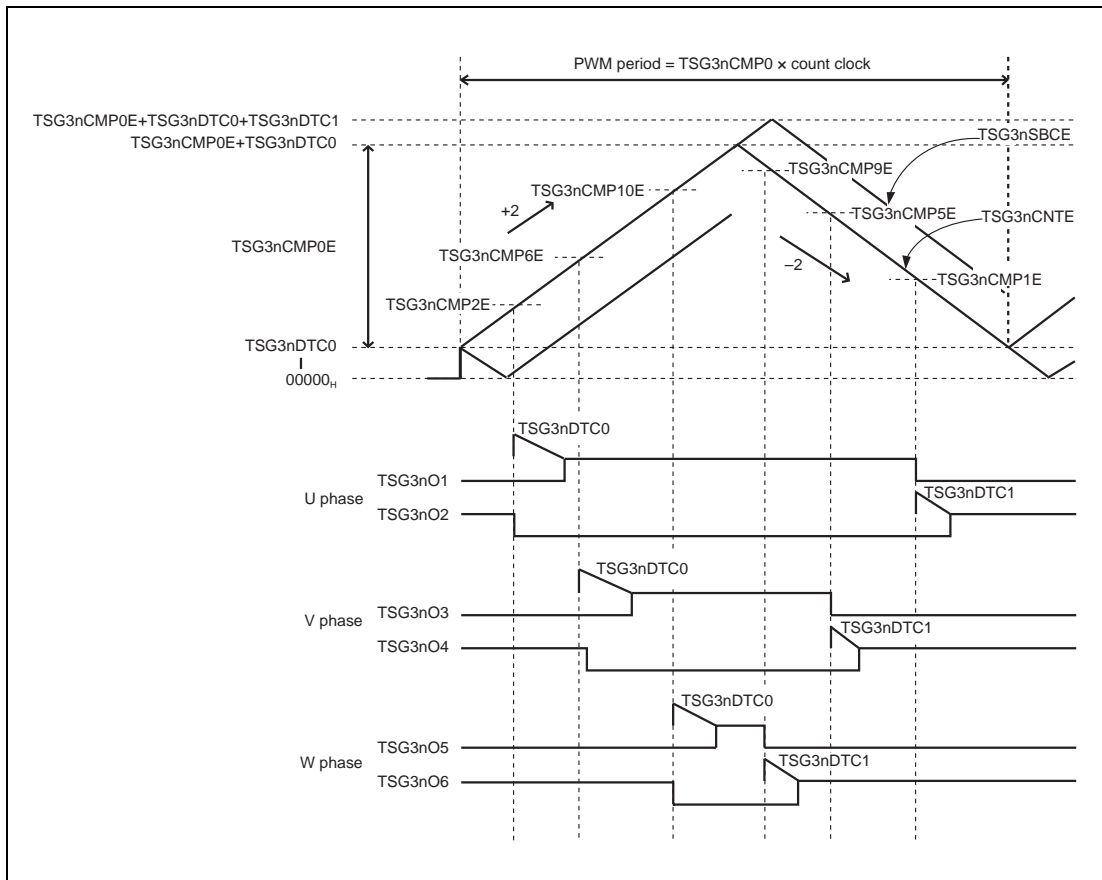


Figure 25.56 Basic Timing in HT-PWM Mode

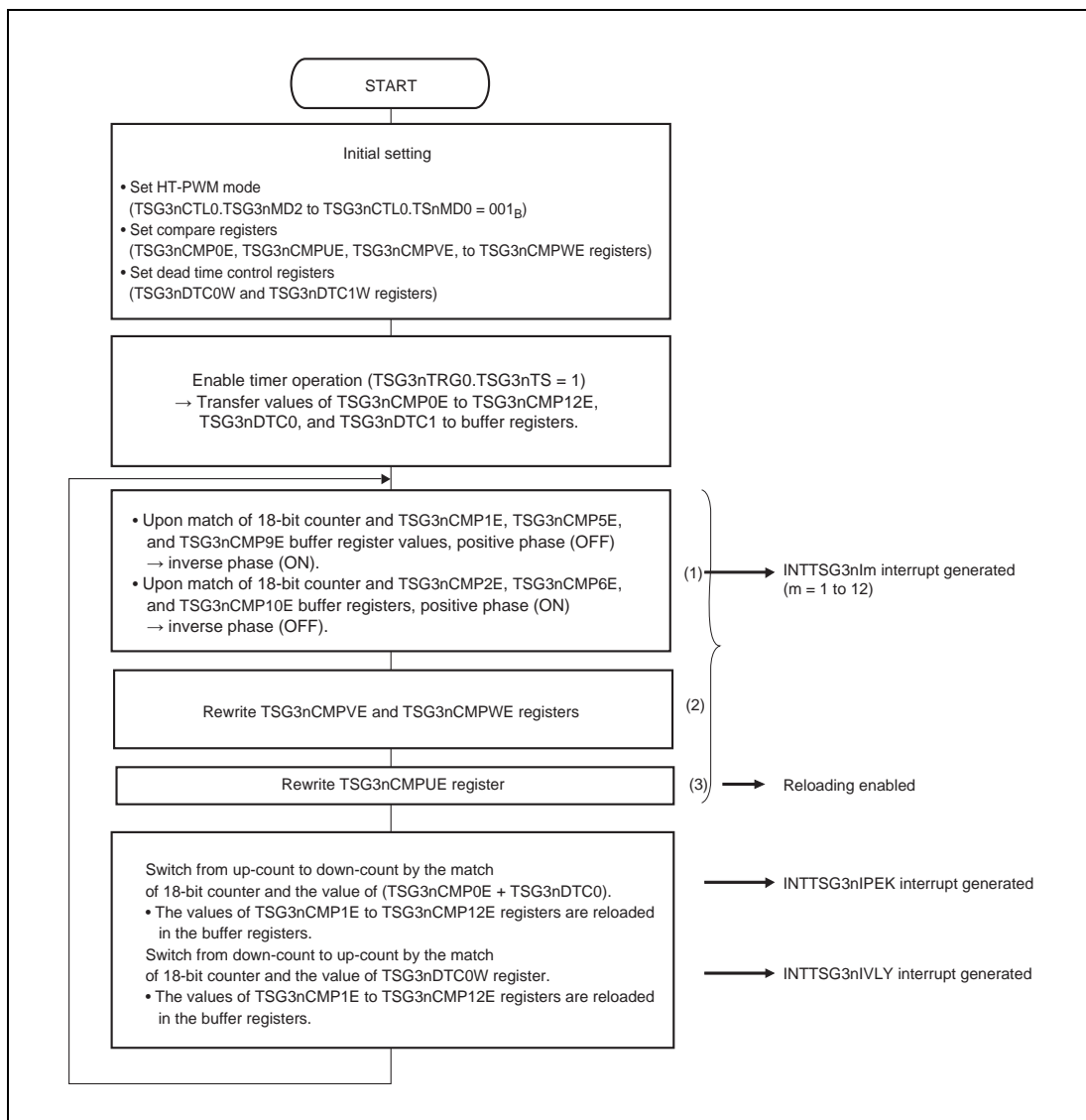


Figure 25.57 Basic Operation Flow in HT-PWM Mode

NOTE

- Write access to TSG3nCMPUE (TSG3nCMP1E) includes reloading enabling operation. Therefore, (3) must be done after (2).
- The INTTSG3nIPEK interrupt is generated only when TSG3nCTL4.TSG3nPIE = 1.
- The INTTSG3nIVLY interrupt is generated only when TSG3nCTL4.TSG3nVIE = 1.
- INTTSG3nI3, INTTSG3nI7, and INTTSG3nI11 outputs an interrupt at the match timing of TSG3nCMP3E, 7E, and 11E with TSG3nCnTE when counting down (TSG3nCnUF = 1). INTTSG3nI4, INTTSG3nI8, and INTTSG3nI12 outputs an interrupt at the match timing of TSG3nCMP4E, 8E, and 12E with TSG3nCnTE when counting up (TSG3nCnUF = 0).

(2) List of HT-PWM Mode Operations

Table 25.61 Counter Function in HT-PWM Mode

| Operation | | Setting Condition |
|--------------------|------------|--|
| 18-bit counter | Start | TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger (counting up from TSG3nDTC0) |
| | Up count | Compare match of TSG3nDTC0 buffer register and 18-bit counter |
| | Down count | Compare match of TSG3nCMP0E + TSG3nDTC0 and 18-bit counter |
| | Clear | — |
| | Stop | TSG3nTRG1.TSG3nTT = 0 → 1 |
| 18-bit sub-counter | Start | TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger (counting down from TSG3nDTC0) |
| | Up count | Underflow |
| | Down count | Compare match of TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 buffer register and 18-bit sub-counter |
| | Load | <ul style="list-style-type: none"> TSG3nCMP0E + TSG3nDTC0: When value of 18-bit counter matches the value of buffer register TSG3nCMP0E + TSG3nDTC0 TSG3nDTC0: When value of 18-bit counter matches the value of the buffer register TSG3nDTC0 |
| | Clear | — |
| | Stop | TSG3nTRG1.TSG3nTT = 0 → 1 |

Table 25.62 Compare Register and Dead Time Control Register Functions in HT-PWM Mode

| Register | Rewrite Method | Rewrite during Operation | Function |
|---------------------------------------|------------------------|--------------------------|--|
| TSG3nCMP0E | Reload/Anytime rewrite | Possible | Setting period |
| TSG3nCMPUE | — | Possible | PWM control for U phase |
| TSG3nCMP1E, TSG3nCMP2E | Reload/Anytime rewrite | | |
| TSG3nCMPVE | — | Possible | PWM control for V phase |
| TSG3nCMP5E, TSG3nCMP6E | Reload/Anytime rewrite | | |
| TSG3nCMPWE | — | Possible | PWM control for W phase |
| TSG3nCMP9E, TSG3nCMP10E | Reload/Anytime rewrite | | |
| TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E | Reload/Anytime rewrite | Possible | Diagnostic signal output or A/D conversion trigger |
| TSG3nDTC0W, TSG3nDTC1W | Reload | Possible conditionally | Period and dead time control |

NOTE

- The rewritten values of TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE are set to TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E and TSG3nCMP10E.
- For rewriting method of TSG3nDTC0 and TSG3nDTC1, see **Section 25.4.7.2, (8), (a) TSG3nDTC0 and TSG3nDTC1 Rewriting.**

Table 25.63 Timer Output Function in HT-PWM Mode

| Pin | Function |
|---------|--|
| TSG3nO0 | Active level output during up count, inactive level output at down count of the 18-bit counter/sub-counter |
| TSG3nO1 | PWM output with dead time by compare match of TSG3nCMP1E buffer register and 18-bit counter (down count) and compare match of TSG3nCMP2E buffer register and 18-bit counter (up count) PWM output by compare match of TSG3nCMP1E buffer register and 18-bit sub-counter (during down counting) and compare match of TSG3nCMP2E buffer register and 18-bit sub-counter (during up counting) while TSG3nCMP1E < DTC0 |
| TSG3nO2 | Output negative phase with respect to TSG3nO1 (with dead time) |
| TSG3nO3 | PWM output with dead time by compare match of TSG3nCMP5E buffer register and 18-bit counter (down count) and compare match of TSG3nCMP6E buffer register and 18-bit counter (up count). PWM output by compare match of TSG3nCMP3E buffer register and 18-bit sub-counter (during down counting) and compare match of TSG3nCMP6E buffer register and 18-bit sub-counter (during up counting) while TSG3nCMP5E < DTC0 |
| TSG3nO4 | Output negative phase with respect to TSG3nO3 (with dead time) |
| TSG3nO5 | PWM output with dead time by compare match of TSG3nCMP9E buffer register and 18-bit counter (down count) and TSG3nCMP10E buffer register and 18-bit counter (up count). PWM output by compare match of TSG3nCMP5E buffer register and 18-bit sub-counter (during down counting) and compare match of TSG3nCMP10E buffer register and 18-bit sub-counter (during up counting) while TSG3nCMP9E < DTC0 |
| TSG3nO6 | Output negative phase with respect to TSG3nO5 (with dead time) |
| TSG3nO7 | Diagnostic signal output or pulse output by A/D conversion trigger |

NOTES

1. The target of TSG3nO0 status output can be switched with TSG3nIOC1.TSG3nTOS.
2. When the peak and valley values of the 18-bit sub-counter are set in TSG3nCMP1E and TSG3nCMP2E, clearing takes precedence.

Table 25.64 Interrupt Request in HT-PWM Mode

| Interrupt | Function |
|---------------------------------------|---|
| INTTSG3nI0 | Compare match of TSG3nDTC0 buffer register and 18-bit counter (periodic interrupt) |
| INTTSG3nIm (m = 1, 2, 5, 6, 9, 10) | Compare match of TSG3nCMPmE buffer register and 18-bit counter (m = 1, 2, 5, 6, 9, 10) |
| INTTSG3nIER | Error interrupt |
| INTTSG3nIVLY | Valley interrupt |
| INTTSG3nIPEK | Peak interrupt |
| INTTSG3nIWN | Warning interrupt |

Table 25.65 Compare Match Timing in HT-PWM Mode

| Compare Match | Timing |
|---------------------------------------|--|
| TSG3nCMP0E | When the 18-bit counter changes from TSG3nDTC0 to TSG3nDTC0 + 2 |
| TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10) | When 18-bit counter changes from TSG3nCMPmE to TSG3nCMPmE ± 2 (m = 1, 2, 5, 6, 9, 10) |

Table 25.66 Example of Setting Each Timer Output Condition in HT-PWM Mode

| Pin | Item | Output Period | Output Duty | |
|---------------------------------|--|--------------------------|--|---|
| | | | Output Condition | Setting Condition |
| TSG3nO0 | Toggle output | TSG3nCMP0E x count clock | Output an inactive level when counting up, and an active level when counting down. | — |
| TSG3nO1, TSG3nO3, TSG3nO5 | PWM output | TSG3nCMP0E x count clock | Output an inactive level throughout one period (0% duty) | $TSG3nCMP0E \leq TSG3nCMPmE \leq TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1$ (m = U, V, W) |
| | | | Output an active level of one count clock in one period | $TSG3nCMPmE = TSG3nCMP0E - 1$ (m = U, V, W) |
| | | | Output an inactive level of one count clock in one period | $TSG3nCMPmE = 0001_H$ (m = U, V, W) |
| | | | Output an active level throughout one period (100% duty) | $TSG3nCMPmE = 0000_H$ (m = U, V, W) |
| TSG3nO2, TSG3nO4, TSG3nO6 | PWM output | TSG3nCMP0E x count clock | Output an inactive level throughout one period (0% duty) | $TSG3nCMPmE \leq TSG3nDTC0 + TSG3nDTC1$ (m = U, V, W) |
| | | | Output an active level of one count clock in one period | $TSG3nCMPmE = TSG3nDTC0 + TSG3nDTC1 + 1$ (m = U, V, W) |
| | | | Output an inactive level of one count clock in one period | $TSG3nCMPmE = TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 - 1$ (m = U, V, W) |
| | | | Output an active level throughout one period (100% duty) | $TSG3nCMPmE = TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1$ (m = U, V, W) |
| TSG3nO7 | Diagnostic signal output or pulse output by A/D conversion trigger | TSG3nCMP0E x count clock | See Section 25.4.5, A/D Conversion Trigger Function | |

(3) Various Settings of HT-PWM Mode

Setting mode

HT-PWM mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 001_B.

Setting timer output

The output pins TSG3nO1 to TSG3nO6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.

The output pin TSG3nO0 indicates the up/down count status of the 18-bit counter or the 18-bit sub-counter. Switch between the 18-bit sub-counter and the 18-bit counter is done with the TSG3nIOC1.TSG3nTOS bit.

The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

Error interrupt (INTTSG3nIER) due to the detection of the simultaneous active state of the positive phase and inverse phase is enabled by setting TSG3nIOC1.TSG3nEOC to 1. In HT-PWM mode, no value specified in the compare register makes the positive phase and negative phase simultaneously active. For the detail, see **Section 25.4.6, Error/Warning Interrupt**.

Setting register rewriting timing with reload function

With TSG3nCTL3.TSG3nRMC, reload (simultaneous rewrite) or rewrite (anytime) is specified for the registers with reload function. The default setting is 0 (reload). To reload, set TSG3nCTL4.TSG3nPRE or TSG3nVRE to 1.

The reload timing is not generated if both TSG3nPRE bit and TSG3nVRE bit are set to 0.

When “anytime rewrite” is specified, unintended output may be generated depending on the rewrite timing.

Setting interrupts and skipping function

Interrupts and the skipping function are set with TSG3nCTL4. TSG3nPIE should be set to 1 when peak interrupt (INTTSG3nIPEK) is required and TSG3nVIE to 1 when valley interrupt (INTTSG3nIVLY) is required. To use the skipping function for peak/valley interrupts, set TSG3nRCC4 to TSG3nRCC0.

Setting A/D conversion trigger output

To set A/D conversion trigger 0 (TSG3nADTRG0 signal), use TSG3nCTL5.TSG3nAT09 to TSG3nAT00. With TSG3nAT09 to TSG3nAT00, A/D conversion trigger output is enabled or disabled at the match of 18-bit counter (during up count) with TSG3nDCMP2E to TSG3nDCMP0E, the match of the 18-bit counter (during down count) with TSG3nDCMP2E to TSG3nDCMP0E, the 18-bit counter peak interrupt (INTTSG3nIPEK), the 18-bit counter valley interrupt (INTTSG3nIVLY), the 18-bit sub-counter peak timing, and 18-bit sub-counter valley timing.

To set A/D conversion trigger 1 (TSG3nADTRG1 signal), use TSG3nCTL6.TSG3nAT19 to TSG3nAT10.

To set the match timing of 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E, set the compare value to the pertinent register.

The skipping function can be used for TSG3nADTRG0 and the TSG3nADTRG1 signals. Use TSG3nACC01, TSG3nACC00 of TSG3nCTL5, and TSG3nACC10, TSG3nACC11 of TSG3nCTL6 to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E to TSG3nDCMP0E correctly when the timing pulse of the A/D conversion trigger is output to TSG3nO7.

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions:

$$\text{PCLK} \times \text{TSG3nDTC0}$$

$$\text{PCLK} \times \text{TSG3nDTC1}$$

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, TSG3nO5 to the inactive state and a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

TSG3nDTC0 and TSG3nDTC1 can only be set to an even value.

Carrier period

Set the carrier period with TSG3nCMP0E according to the following expression:

$$\text{TSG3nCMP0E} = \text{Carrier period/count clock period (PCLK)}$$

The following requirements regarding the dead time must be satisfied when setting the TSG3nCMP0E register:

- $\text{TSG3nCMP0E} + \text{TSG3nDTC0} + \text{TSG3nDTC1} \leq 3\text{FFFEH}$
- $\text{TSG3nCMP0E} > \text{TSG3nDTC0}$
- $\text{TSG3nCMP0E} > \text{TSG3nDTC1}$
- $\text{TSG3nCMP0E} > 3 \times \text{MAX}(\text{TSG3nDTC0}, \text{TSG3nDTC1})$
- TSG3nCMP0E: Even number

NOTE

MAX (A, B) indicates the larger value of A and B.

Setting duty (PWM width)

The duty of the U phase, the V phase, and the W phase is set with TSG3nCMPmE (m = U, V, W, or 1, 2, 5, 6, 9, and 10), respectively. The setting range of the compare registers is as follows:

$$00000_{\text{H}} \leq \text{TSG3nCMPmE} \leq \text{TSG3nCMP0E} + \text{TSG3nDTC0} + \text{TSG3nDTC1}$$

LSB (least significant bit) of TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE indicates the setting of an additional pulse. For example, when $\text{TSG3nCMPUE} = 00003_{\text{H}}$, the change in the inverse phase (TSG3nO2 output) occurs one count clock later compared to the $\text{TSG3nCMPUE} = 00002_{\text{H}}$

setting (when the 18-bit counter is up-counting). The additional pulse cannot be set to TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, or TSG3nCMP10E (only even numbers can be set to these registers).

(4) 18-bit counter Operation in HT-PWM Mode

The 18-bit counter is initialized to 00000_{16} and the value of TSG3nDTC0 is loaded immediately after the TSG3n timer operation starts ($TSG3nTRG0.TSG3nTS = 1$), and the counter is incremented by 2. After 18-bit counter reaches the value of $TSG3nCMP0E + TSG3nDTC0$, it is decremented by 2.

The following figure shows 18-bit counter operation.

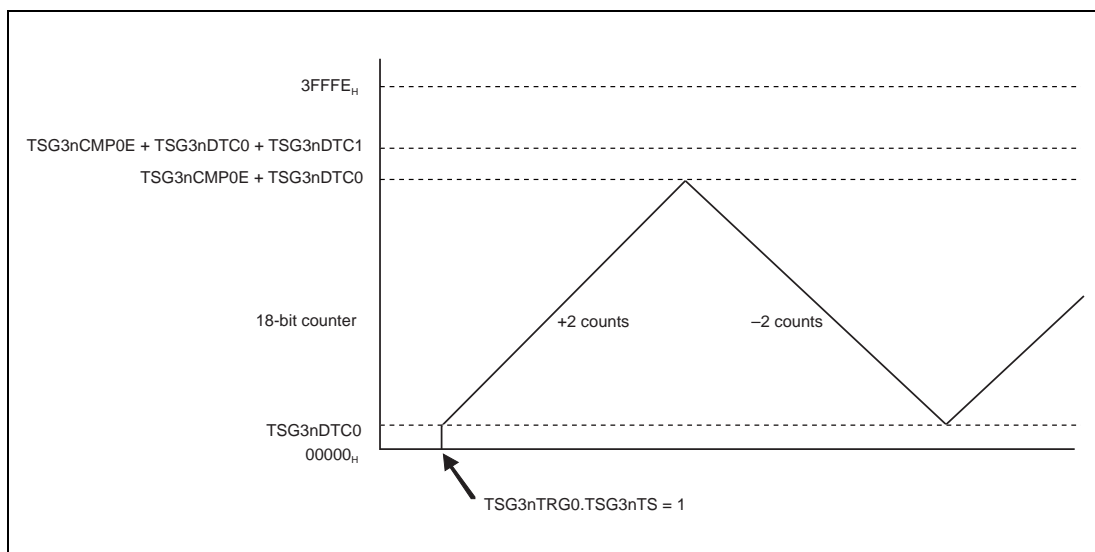


Figure 25.58 Example of 18-bit counter Operation in HT-PWM Mode

NOTE

Minimum 18-bit counter value: TSG3nDTC0

Maximum 18-bit counter value: TSG3nCMP0E + TSG3nDTC0

Carrier period: TSG3nCMP0E × count clock period (PCLK)

The 18-bit sub-counter is initialized to 00000_H and the value of TSG3nDTC0 is loaded immediately after the TSG3n timer operation starts ($TSG3nTRG0.TSG3nTS = 1$), and the counter is decremented by 2 until it reaches 00000_H , at which point increment by 2 begins. Next, the value of the 18-bit counter is loaded into the 18-bit sub-counter at a change timing of the 18-bit counter into the down count. Counting up by the 18-bit sub-counter continues until the value reaches the value of $TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1$, and then decrement by 2 begins. Similarly, when the 18-bit counter value matches the TSG3nDTC0 value, the 18-bit counter value is loaded to the 18-bit sub-counter and the down count is continued.

The following figure shows the 18-bit sub-counter operation.

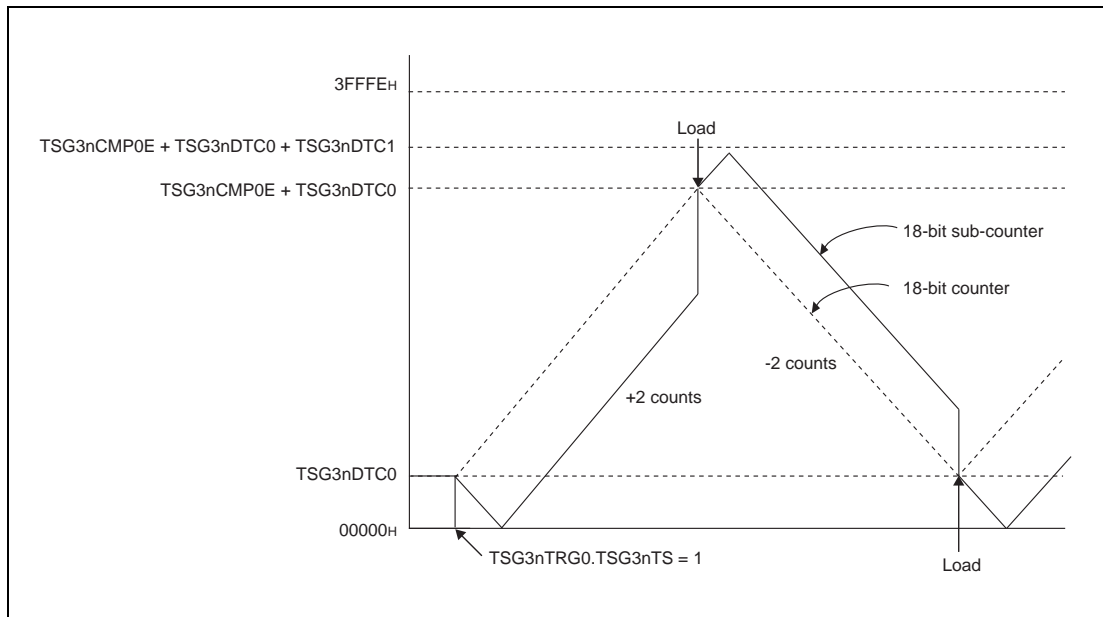


Figure 25.59 Example of 18-bit Sub-Counter Operation in HT-PWM Mode

NOTE

Minimum 18-bit sub-counter value: 00000_H

Maximum 18-bit sub-counter value: $TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1$

(5) Basic Operation of HT-PWM Mode

(a) Example of Timer Output Immediately after the Start of the TSG3n Timer Operation

The following figure shows the timing chart when TSG3nCMP0E = 0000E_H, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H and TSG3nCMPUE = 00000_H to 00014_H (excerpt). In this example, TSG3nIOC2.TSG3nOL1 to TSG3nOL6 = 000000_B.

When operation starts (TSG3nTRG0.TSG3nTS = 1), the level of the TSG3nO2 pin changes to active. Afterwards, if TSG3nCMPUE ≤ TSG3nDTC0, the TSG3nO2 pin is cleared after 1 count clock cycle.

The TSG3nO2 pin is cleared upon a match of the 18-bit counter and the compare register (TSG3nCMP2E), or a match of the 18-bit sub-counter and the compare register (TSG3nCMP2E) if TSG3nCMPUE ≥ TSG3nDTC0. Afterwards, the TSG3nO1 pin is set after the set dead time period (the TSG3nO1 pin is not set if TSG3nCMPUE ≥ TSG3nCMP0E).

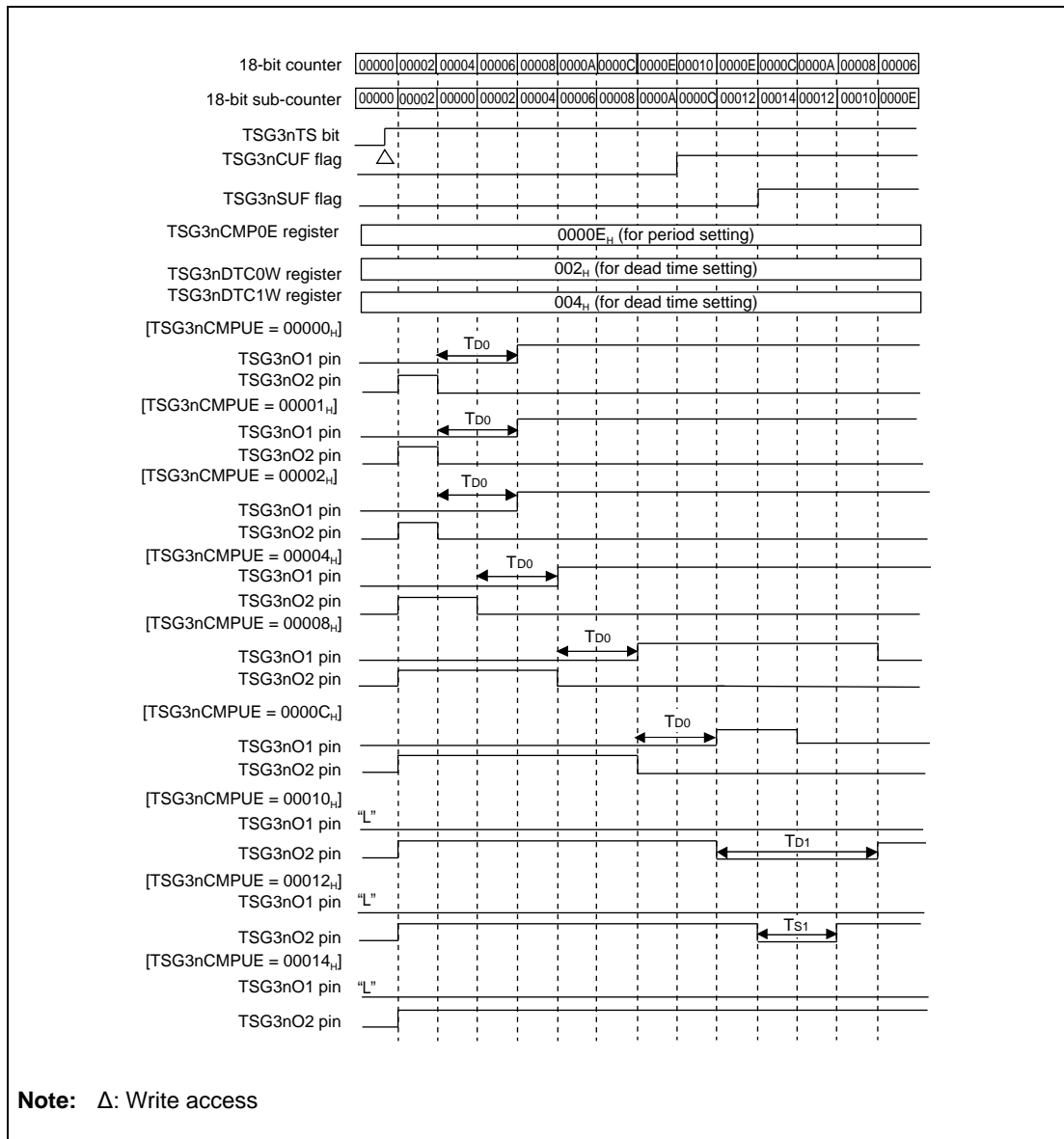


Figure 25.60 Example of Timer Output when TSG3nTS is Set to 1 (Initial Setting) in HT-PWM Mode

NOTES

1. TSG3nCMP0E = 0000E_H, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H
 2. T_{D0}: Time depending on the dead time setting in the TSG3nDTC0W register
T_{D1}: Time depending on the dead time setting in the TSG3nDTC1W register
T_{S1}: Time determined by compare match of the 18-bit sub-counter and TSG3nCMPUE register, when TSG3nCMPUE > 18-bit counter maximum value
-

(b) Example of Timer Output during TSG3n Timer Operation

The following figure shows the timing chart when TSG3nCMP0E = 0000E_H, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H, and TSG3nCMPUE is set to 00000_H-00014_H (excerpt). In this example, TSG3nIOC2.TSG3nOL1-TSG3nOL6 = 000000_B.

The range of the active (high level) width of a positive phase (TSG3nO1) output is 00000_H ≤ TSG3nCMPUE ≤ TSG3nCMP0E (for the additional pulse). The range of the active (high level) width of an inverse phase (TSG3nO2) output is TSG3nDTC0 + TSG3nDTC1 ≤ TSG3nCMPUE ≤ TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1.

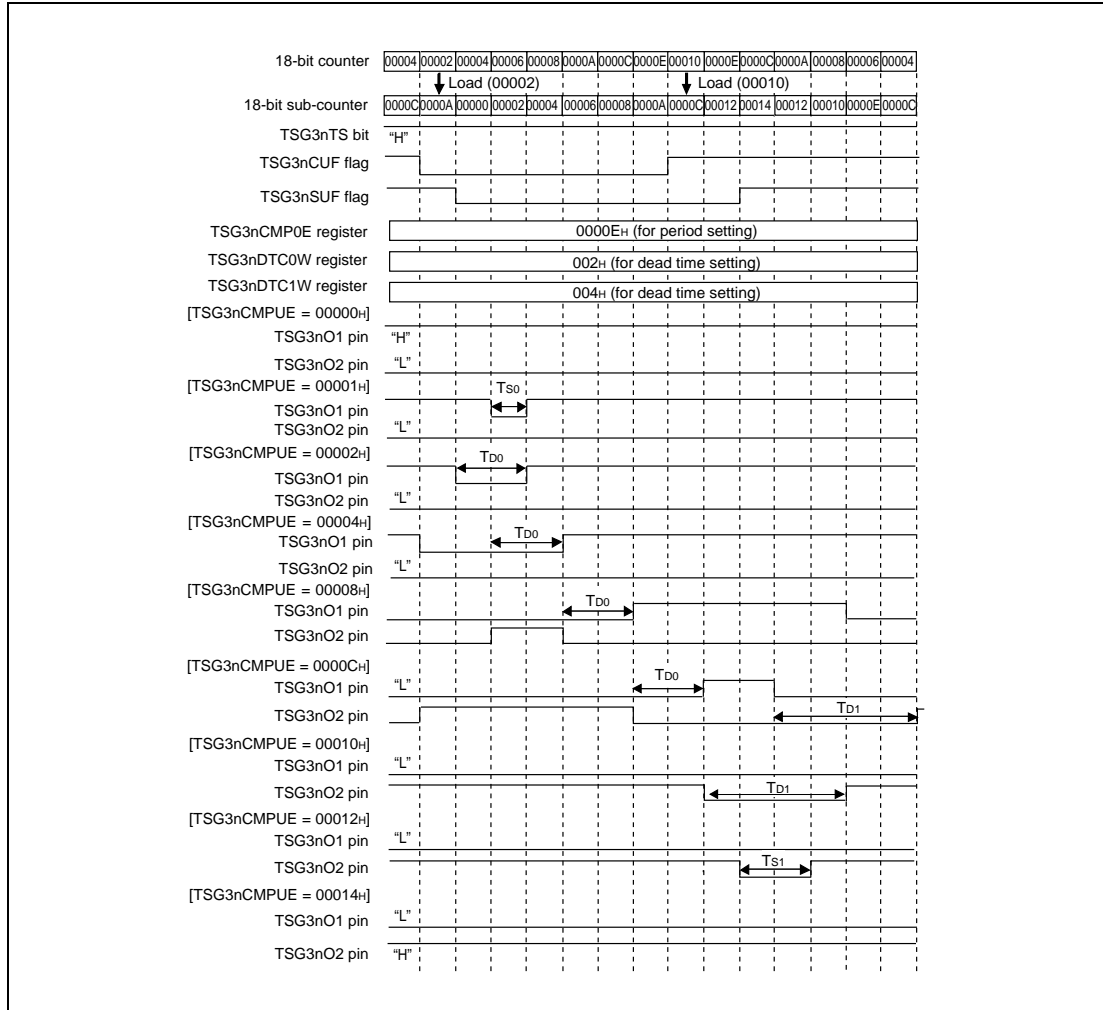


Figure 25.61 Example of Timer Output during TSG3n Operation in HT-PWM Mode

NOTES

1. TSG3nCMP0E = 0000E_H, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H
2. T_{DO}: Time depending on setting of the dead time in the TSG3nDTC0 register
 T_{D1}: Time depending on setting of the dead time in the TSG3nDTC1 register
 T_{S0}: Time determined by compare match of 18-bit sub-counter and the TSG3nCMPUE register, when TSG3nCMPUE < 18-bit counter minimum value
 T_{S1}: Time determined by compare match of the 18-bit sub-counter and the TSG3nCMPUE register, when TSG3nCMPUE > 18-bit counter maximum value

(6) Additional Pulse Control in HT-PWM Mode

The HT-PWM mode can generate an additional pulse by setting 1 to the LSB of the duty setting registers (TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE). This allows more precise control of the pulse duty than standard pulse control.

The following sections describe two examples of pulse output of TSG3nO1: additional pulse control is used in one example and additional pulse control is not used in another.

(a) Example of Pulse Output when Additional Pulse Control Is Used

Figure 25.62 shows the additional pulse control when an odd value is set to TSG3nCMPUE.

The arrows and numerical values show the duty cycle width of the TSG3nO1 output in one period.

As shown in **Figure 25.62**, when the additional pulse control is used, the output width of the TSG3nO1 (duty cycle) can be set within a range from 12 clock cycles to 0 clock cycles in one-clock-cycle step.

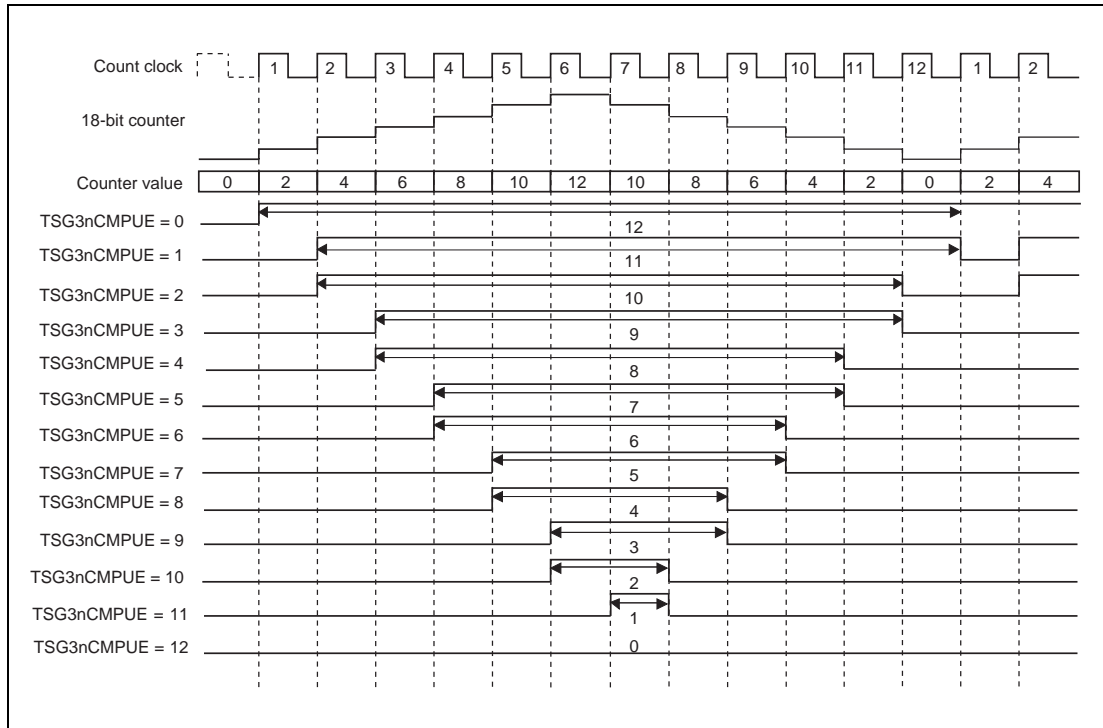


Figure 25.62 Example of TSG3nO1 Output when Additional Pulse Control Is Used in HT-PWM Mode

NOTE

TSG3nCMP0E = 12, TSG3nDTC0 = 0, TSG3nDTC1 = 0

(b) Example of Pulse Output when Additional Pulse Control Is Not Used

The arrows and numerical values in **Figure 25.63** show the duty cycle width of the TSG3nO1 output in one period.

When the additional pulse control is not used, the width of the TSG3nO1 output can be set within a range from 12 clock cycles to 0 clock cycles in two-clock-cycle step. In this case, the change in duty cycle is larger than that of the case when the additional pulse control is used.

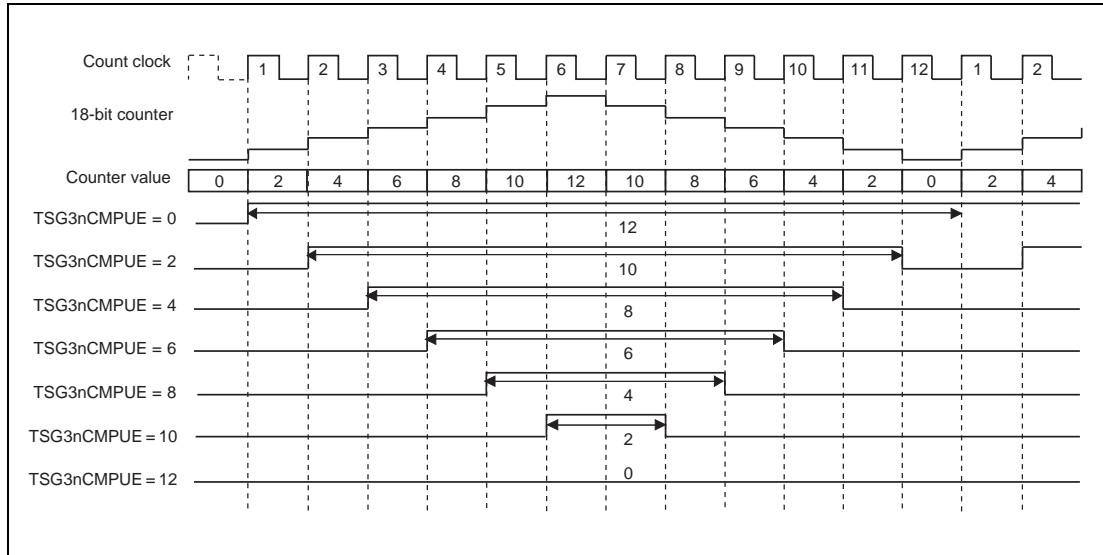


Figure 25.63 Example of TSG3nO1 Output when Additional Pulse Control Is Not Used in HT-PWM Mode

NOTE

TSG3nCMP0E = 12, TSG3nDTC0 = 0, TSG3nDTC1 = 0

(7) Dead Time Control in HT-PWM Mode

Duty setting are basically configured with the TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE registers in HT-PWM mode. The 6-phase PWM waveform of a variable duty is output by using these registers. To achieve the dead time control, there are two dead time control registers (TSG3nDTC0W and TSG3nDTC1W) and six 10-bit down counters that operate synchronously with the count clock of the 18-bit counter. TSG3nDTC0 is used for setting a dead time from a change of the inverse phase to the inactive state to a change of the positive phase to the active state. TSG3nDTC1 is used for setting a dead time from a change of the positive phase to the inactive state to a change of the inverse phase to the active state.

The following figure shows the output waveform when TSG3nDTC0 = x and TSG3nDTC1 = y.

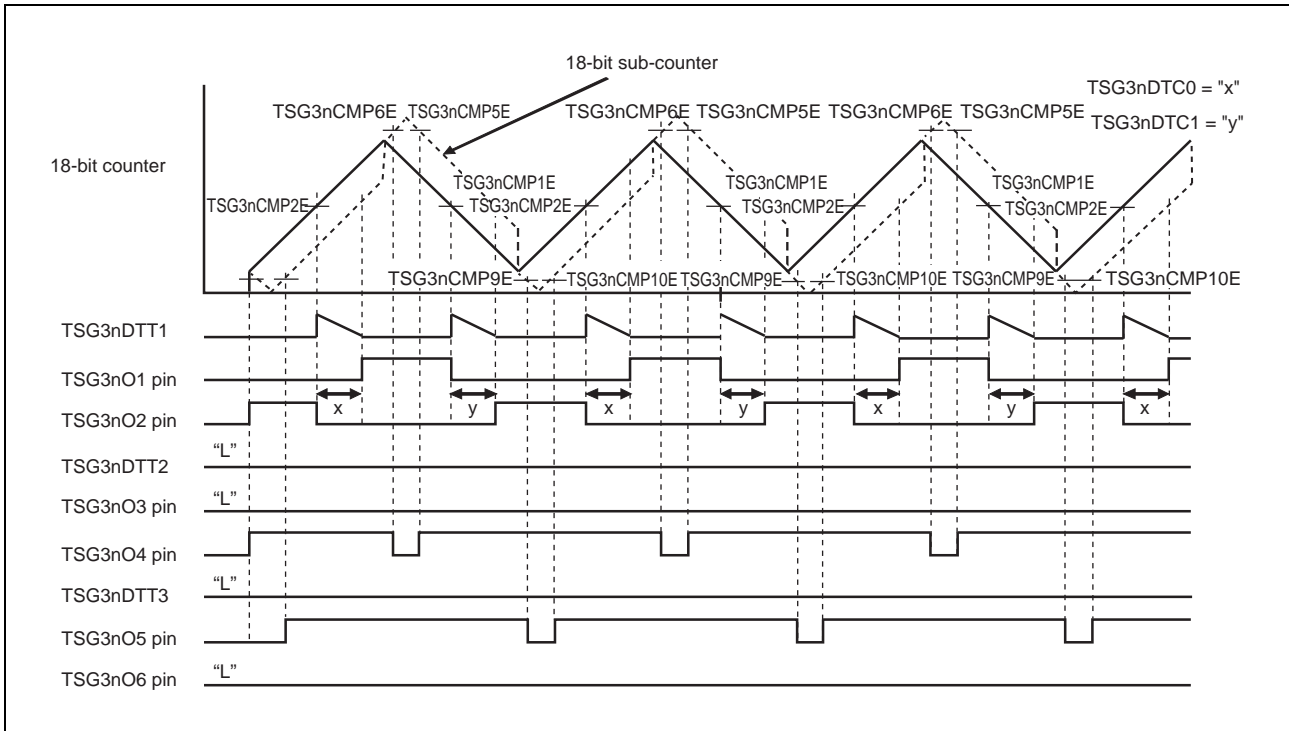
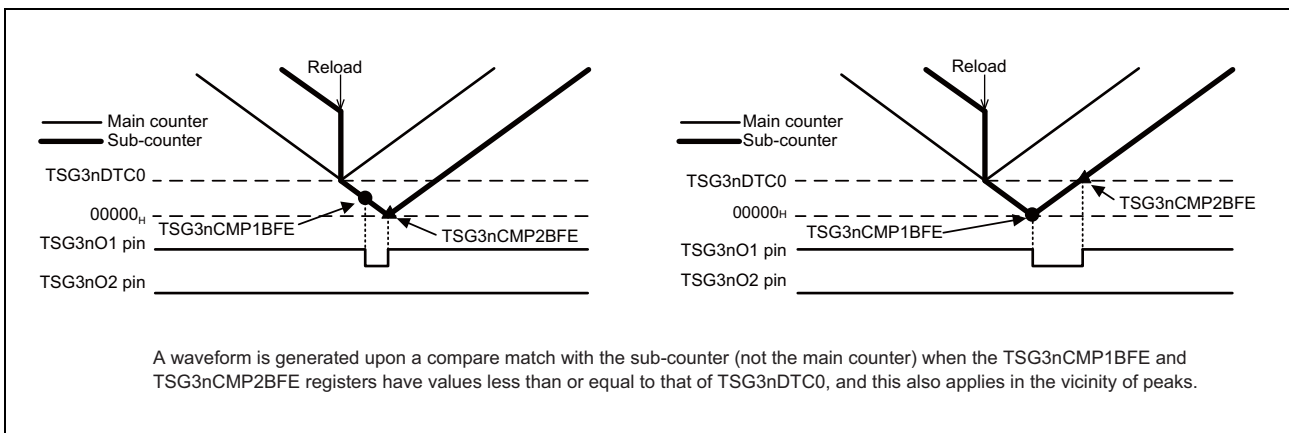


Figure 25.64 Example of Output Waveform with Dead Time in HT-PWM Mode



A waveform is generated upon a compare match with the sub-counter (not the main counter) when the TSG3nCMP1BFE and TSG3nCMP2BFE registers have values less than or equal to that of TSG3nDTC0, and this also applies in the vicinity of peaks.

Figure 25.65 Example of Output Waveform near Trough after Reloading

(8) Notes Concerning Dead Time Control in HT-PWM Mode

(a) TSG3nDTC0 and TSG3nDTC1 Rewriting

It is possible to rewrite the dead time setting in TSG3nDTC0 and TSG3nDTC1 registers during timer operation.

CAUTIONS

1. Rewrite TSG3nDTC0 and TSG3nDTC1 when the reload function is used (TSG3nRMC = 0).
2. The write protection code check function is applied when TSG3nDTC0 and TSG3nDTC1 are rewritten. For details, see the pertinent register descriptions (Section 25.3.43, Section 25.3.44, Section 25.3.63).
3. When the TSG3nCMP0E and Tsg3nDTC1 are updated at the peak of the 18-bit counter:
 If the set value of TSG3nCMPmE is greater than the updated TSG3nCMP0E + TSG3nDTC0 value (new maximum value of the main counter), the match interrupt (INTTSG3nlm) is not generated immediately after reloading (m = 2, 6, or 10).

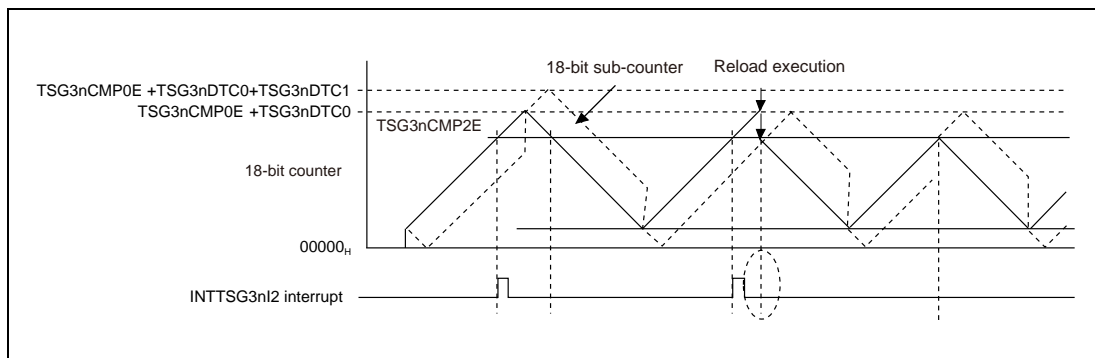


Figure 25.66 Example of Operation During Reloading at 18-Bit Counter Peak Timing

4. When the TSG3nDTC0 is updated at the valley of the 18-bit counter:
 If the TSG3nCMPmE set value is smaller than the updated TSG3nDTC0 value (new minimum value of the main counter), the match interrupt (INTTSG3nlm) is not generated immediately after reloading (m = 1, 5, or 9).

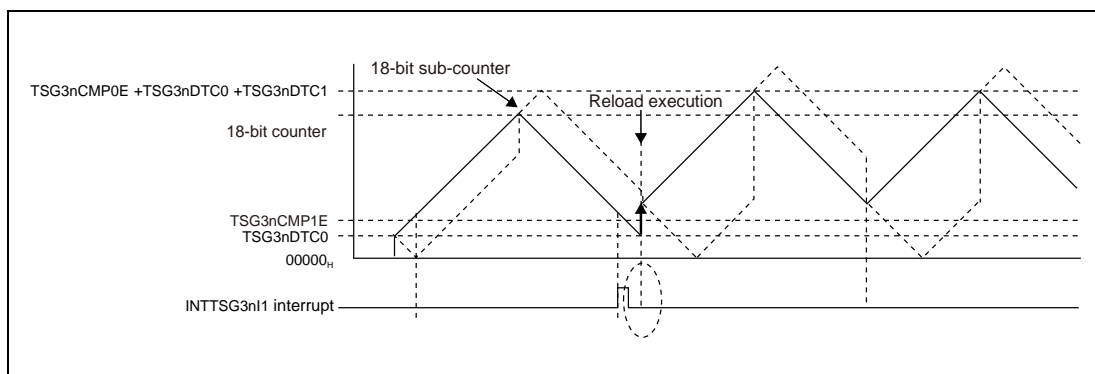


Figure 25.67 Example of Operation During Reloading at 18-Bit Counter Trough Timing

(9) Software Output Control Function in HT-PWM Mode

TSG3nOPT0.TSG3nSOC, TSG3nIDC, and TSG3nOPT1.TSG3nSPC2-TSG3nSPC0 are used in HT-PWM mode for software control of timer output control.

As shown in **Figure 25.68**, with TSG3nSTE = 0, the output control is switched immediately when TSG3nSOC is set to 1. If the dead time is set, the period of the dead time is guaranteed. After that, when TSG3nSOC is set to 0, output control is retained and at the reload timing, output control is switched to HT-PWM mode output control.

For details, refer to **Section 25.4.7.8, Software Output Control Function**.

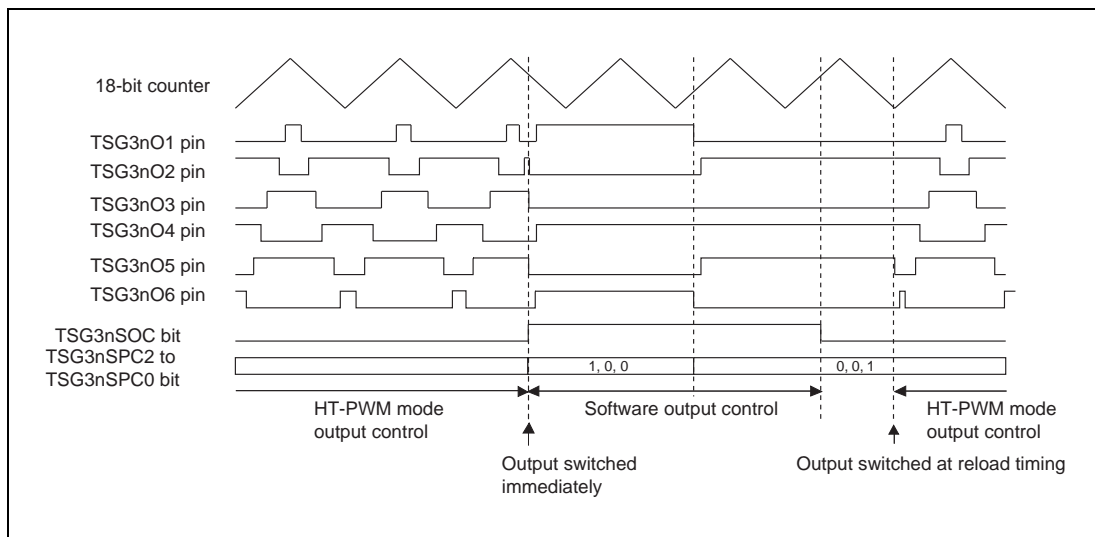


Figure 25.68 Example of Switching from HT-PWM Mode Control to Software Output Control

CAUTION

Use reload (simultaneous rewrite) mode (TSG3nCTL3.TSG3nRMC = 0) when software output control function is used.

(a) Procedure for Software Output Control

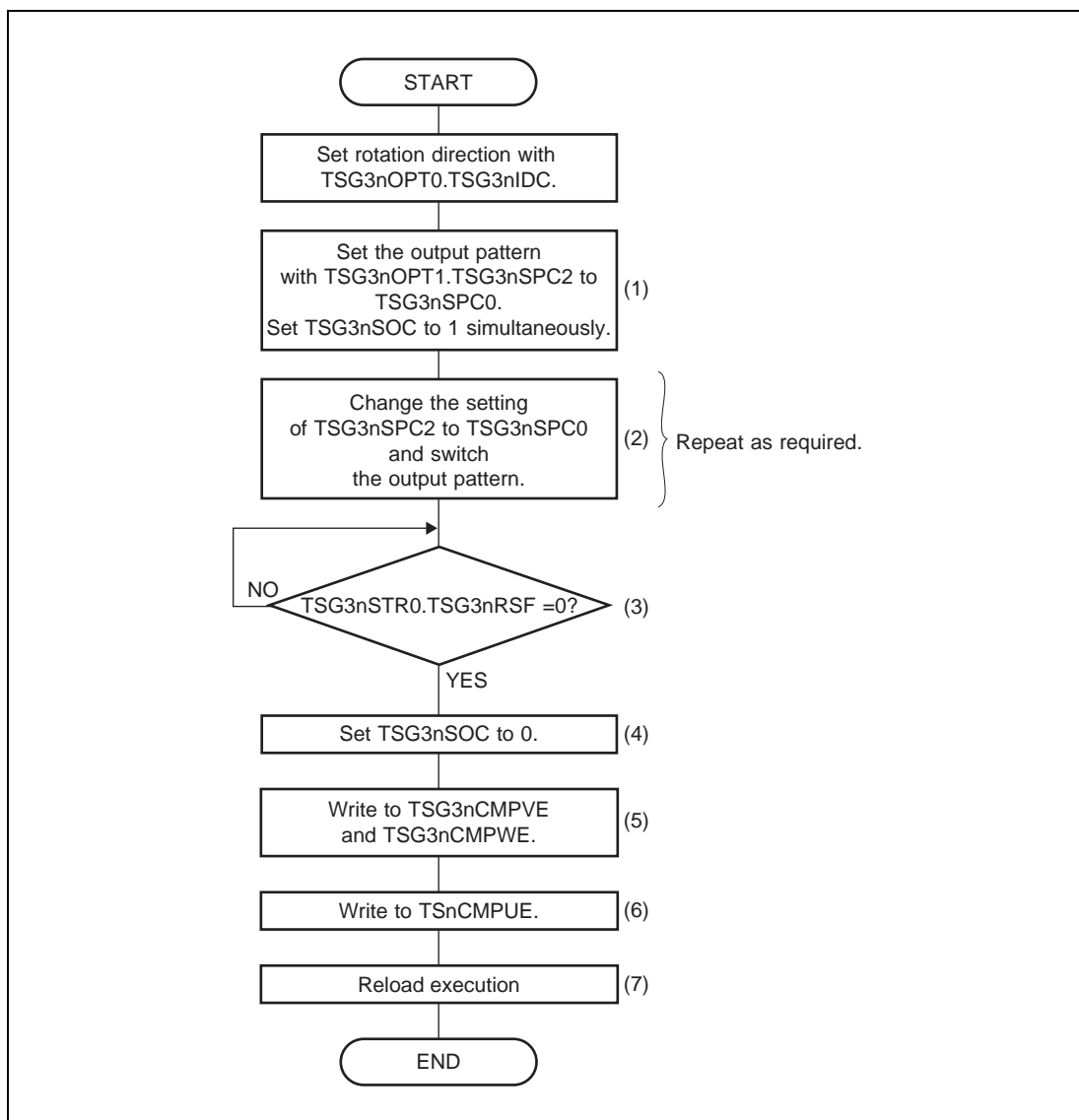


Figure 25.69 Flow of Software Output Control in HT-PWM Mode

The procedure for software output control is described below.

- (1) Set the output pattern to the TSG3nOPT1.TSG3nSPC2-TSG3nSPC0. To enable software output control, set TSG3nOPT0.TSG3nSOC to 1 simultaneously.
- (2) Change the output pattern setting of the TSG3nSPC2-TSG3nSPC0 to change the timer output.
- (3) Confirm that reload request flag TSG3nSTR0.TSG3nRSF = 0. In case TSG3nRSF = 1, do not proceed to the following step until TSG3nRSF = 0.
- (4) By setting TSG3nSOC = 0, the software control release process starts (it is not released yet at this point).
- (5) Set the compare register settings that are required after the software output control is released. Proceed to the following step when the register setting is not required. When configuring the registers with the reload function, set them at this point.
- (6) Write to TSG3nCMPUE (TSG3nCMP1E) to start reloading.
- (7) Reload is executed and software output is released.

CAUTION

Execute reload after completing steps (3), (4), (5), and (6). When reload cannot be executed, the software output cannot be released.

(10) Asymmetric Triangular Wave Control in HT-PWM Mode

In HT-PWM mode, it is possible to control output by an asymmetric triangular waveform by setting the different timings for setting and clearing the U, V, and W phases.

The following describes the differences of the asymmetric triangular wave control from the symmetric triangular wave control.

(a) PWM Setting

When a symmetric triangular wave is used, the output control of each of the U phase, V phase, and W phase is done by setting the set timing and the clear timing to the same value in TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE, respectively. When an asymmetric triangular wave is used, the output control of each phase is done by setting TSG3nCMPmE as follows ($m = 1, 2, 5, 6, 9, 10$).

Prerequisites

- The clear timing of PWM of the voltage data signal of U, V and W phases is set with TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E.
- The set timing of PWM of the voltage data signal of U, V, and W phases is set with TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E.
- The set and clear timings of each phase can also be set with TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E.
- TSG3nCMPmE can only be set to an even value ($m = 1, 2, 5, 6, 9, 10$).

(b) Timer Output

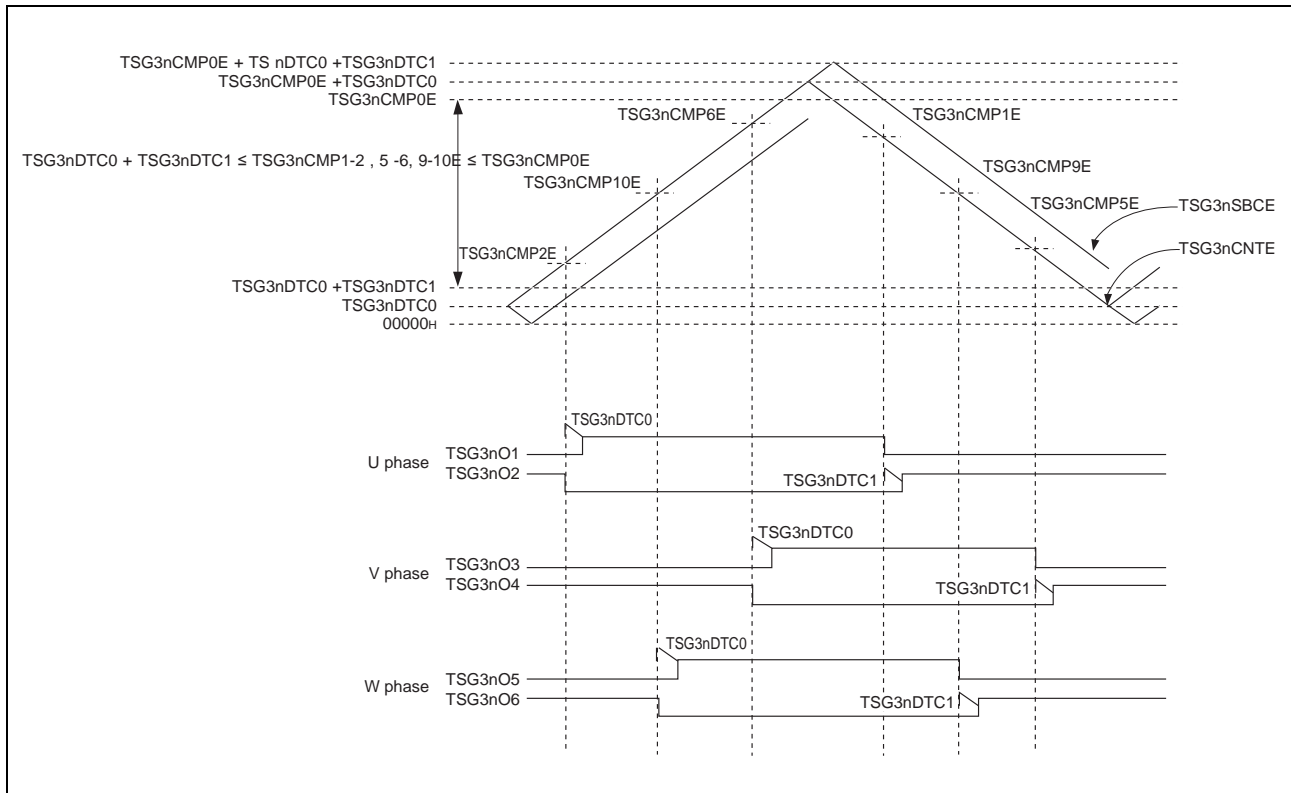


Figure 25.70 Example of Timer Output Waveform in HT-PWM Mode

NOTE

When output is controlled by the asymmetric triangular waveform, the following conditions apply to setting of $TSG3nCMPmE$ ($m = 1, 2, 5, 6, 9, 10$).

- $TSG3nDTC0 + TSG3nDTC1 \leq TSG3nCMPmE \leq TSG3nCMP0E$
- Only when $TSG3nCMPmE = TSG3nCMP(m + 1)E$, or $TSG3nCMPmE = TSG3nCMP(m + 1)E + 2$, it is possible to set $TSG3nCMPmE$ under the condition $00000_H \leq TSG3nCMPmE \leq TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1$, which also applies to the case where the symmetric triangular wave is used.

25.4.7.3 SP-PWM Mode (Shifted-pulse - Pulse Width Modulation Mode)

Overview

In this mode, a 6-phase PWM can be generated using the 18-bit counter and the 18-bit compare registers.

Prerequisites

- The PWM signal cycle is set in TSG3nCMP0E.
- The set timings of the U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E, while the clear timings of these phases are configured using TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E (when set timing and clear timing are used for control).
- The set timings of the U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E while the active periods of these phases are configured using TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE. The sum of the set values of TSG3nCMP2E, TSG3nCMP6E, TSG3nCMP10E, and the set values of TSG3nUPWE, TSG3nVPWE, TSG3nWPWE are set to TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E respectively (when set timing and active period are used for control). The value after addition must be no greater than 3FFFF_H. Otherwise, values having 19 or more bits are truncated.

Functional description

In this mode, the carrier period, the set timings, and the duty cycle of the U phase, V phase, and W phase are configured. The counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter counts up from 00000_H and is cleared by match with TSG3nCMP0E.

The dead time is set with TSG3nDTC0 and TSG3nDTC1. TSG3nDTC0 sets the dead time of inverse phase (OFF) to positive phase (ON) switch while TSG3nDTC1 sets that of positive phase (OFF) to inverse phase (ON) switch. The 10-bit counters (TSG3nDTT1 to TSG3nDTT3) for dead time generation load the set values of TSG3nDTC0 and TSG3nDTC1 at compare match of the 18-bit counter with the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) and start down-counting.

INTTSG3nIm interrupts (m = 1, 2, 5, 6, 9, 10) are generated by the compare match of the 18-bit counter with the TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E buffer registers.

NOTE

SP-PWM mode is enabled when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 010_B.

(1) Basic Timing Chart

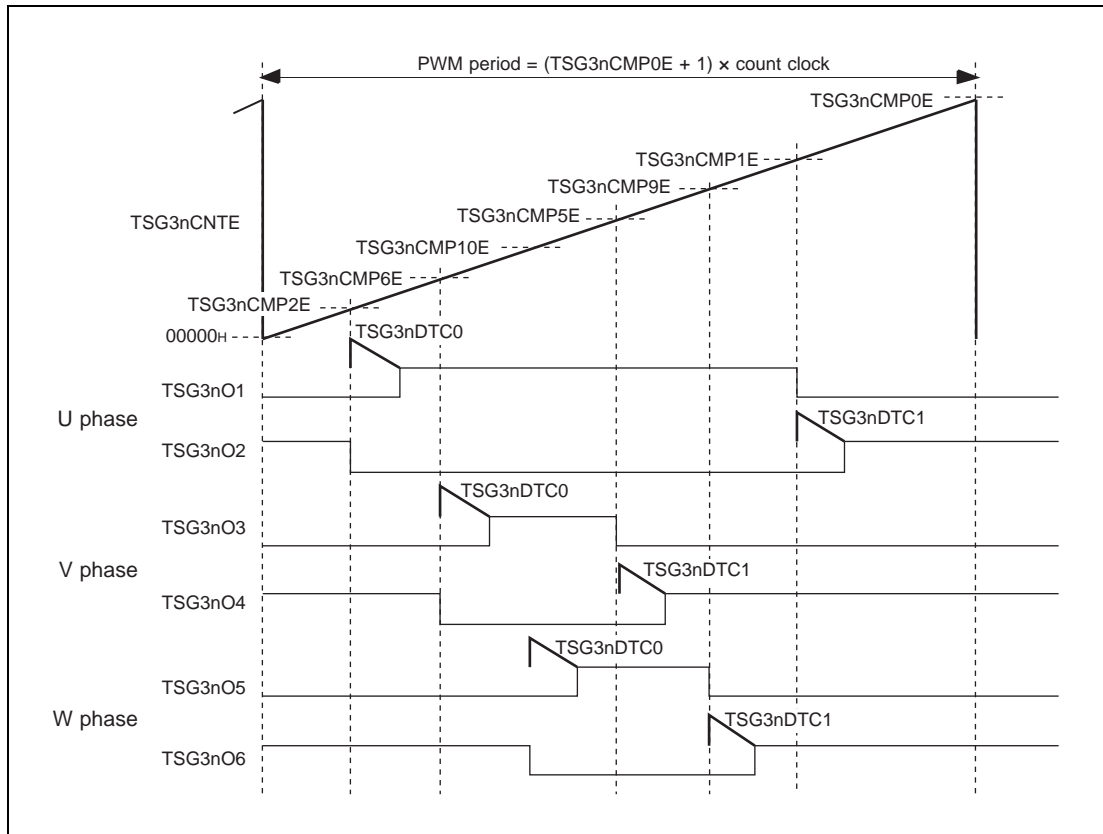


Figure 25.71 Basic Timing in SP-PWM Mode

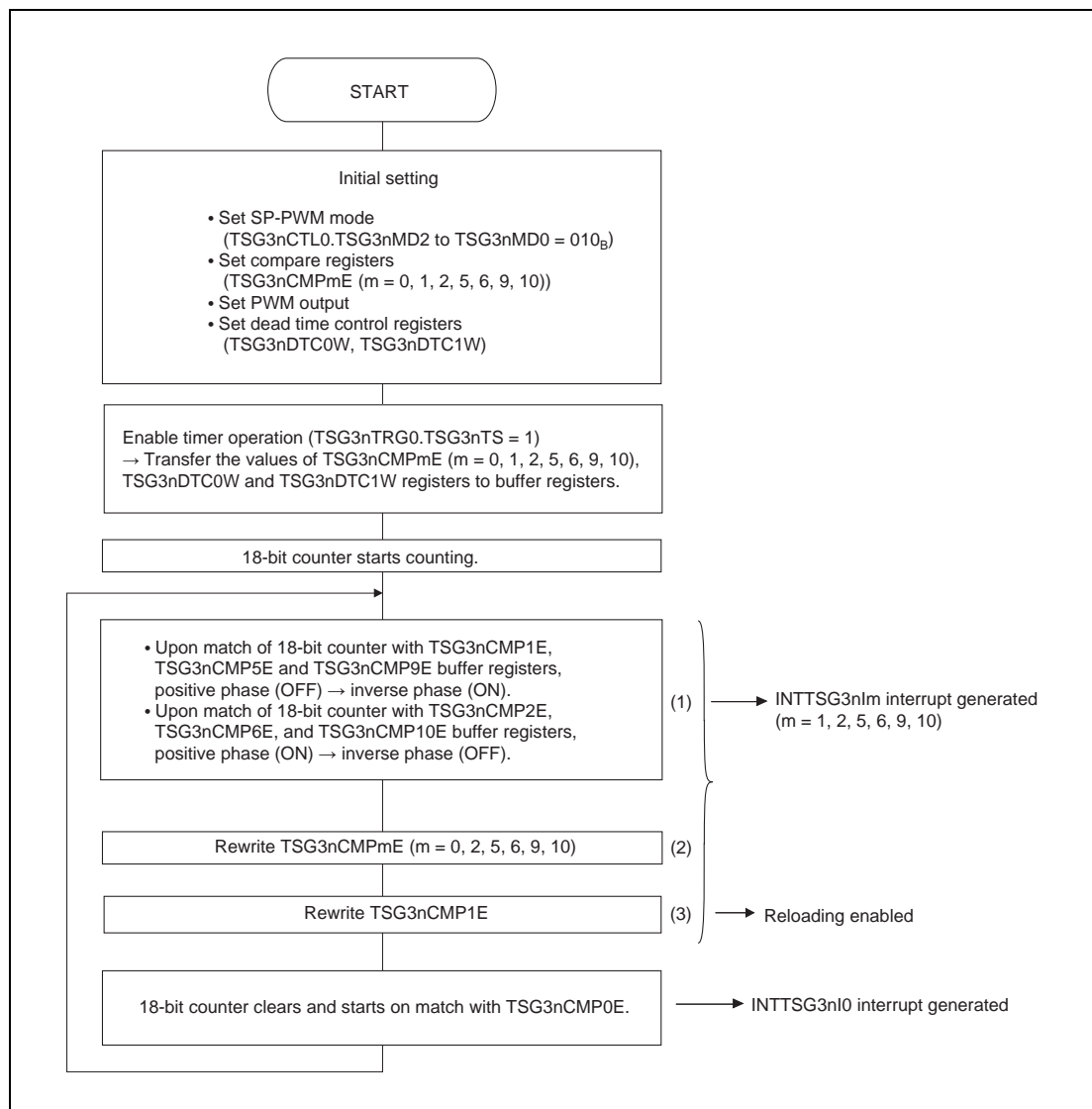


Figure 25.72 Basic Operation Flow in SP-PWM Mode

NOTE

The timing of (1) may be different depending on the rewriting timing of (2) and (3) and the TSG3nCmP1E value. Be sure to rewrite (2) before rewriting (3).

(2) List of SP-PWM Mode Operations

Table 25.67 Counter Functions in SP-PWM Mode

| Operation | | Setting Condition |
|----------------|-------|---|
| 18-bit counter | Start | TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger |
| | Clear | Compare match of TSG3nCMP0E buffer register with 18-bit counter |
| | Stop | TSG3nTRG1.TSG3nTT = 0 → 1 |

Table 25.68 Compare Registers and Dead Time Control Register Functions in SP-PWM Mode

| Register | Rewriting Method | Rewrite during Operation | Function |
|---|------------------------|--------------------------|---|
| TSG3nCMP0E | Reload/Anytime rewrite | Possible | Setting period |
| TSG3nUPWE | Reload/Anytime rewrite | Possible | PWM control for U phase |
| TSG3nCMP1E, TSG3nCMP2E | Reload/Anytime rewrite | | |
| TSG3nVPWE | Reload/Anytime rewrite | Possible | PWM control for V phase |
| TSG3nCMP5E, TSG3nCMP6E | Reload/Anytime rewrite | | |
| TSG3nWPWE, TSG3nCMP9E, TSG3nCMP10E | Reload/Anytime rewrite | Possible | PWM control for W phase |
| TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E | Reload/Anytime rewrite | Possible | Diagnostic output or A/D conversion trigger |
| TSG3nDTC0W, TSG3nDTC1W | Reload | Possible | Period and dead time |

Table 25.69 Timer Input Function in SP-PWM Mode

| Pin | Function |
|-----------|--------------------|
| TSG3nCLKI | Clock enable input |

Table 25.70 Output Functions in SP-PWM Mode

| Pin | Function |
|---------|---|
| TSG3nO1 | PWM output with dead time by compare match of TSG3nCMP1E buffer register (clear timing) or TSG3nCMP2E buffer register (set timing) with 18-bit counter |
| TSG3nO2 | Output inverse phase with respect to TSG3nO1 (with dead time) |
| TSG3nO3 | PWM output with dead time by compare match of TSG3nCMP5E buffer register (clear timing) or TSG3nCMP6E buffer register (set timing) with 18-bit counter |
| TSG3nO4 | Output inverse phase with respect to TSG3nO3 (with dead time) |
| TSG3nO5 | PWM output with dead time by compare match of the TSG3nCMP9E buffer register (clear timing) or the TSG3nCMP10E buffer register (set timing) with the 18-bit counter |
| TSG3nO6 | Output inverse phase with respect to TSG3nO5 (with dead time) |
| TSG3nO7 | Diagnostic signal output or pulse output by A/D conversion trigger |

Table 25.71 Interrupt Requests in SP-PWM Mode

| Interrupt | Function |
|---------------------------------------|--|
| INTTSG3nIm (m = 0, 1, 2, 5, 6, 9, 10) | Compare match of TSG3nCMPmE buffer register with 18-bit counter (m = 0, 1, 2, 5, 6, 9, 10) |
| INTTSG3nIER | Error |
| INTTSG3nIVLY | — |
| INTTSG3nIPEK | Peak interrupt (generated at the same timing as INTTSG3nI0 interrupt) |
| INTTSG3nIWN | Warning |

Table 25.72 Compare Match Timing in SP-PWM Mode

| Compare Match | Timing |
|------------------------------------|--|
| TSG3nCMP0E | When 18-bit counter changes from TSG3nCMP0E to 00000 _H |
| TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10) | After match of 18-bit counter and TSG3nCMPmE is detected (m = 1, 2, 5, 6, 9, 10) |

Table 25.73 Example of Setting Each Timer Output Condition in SP-PWM Mode

| Pin | Item | Output Period | Output Duty | |
|---------------------------------|--|-----------------------------------|---|---|
| | | | Output Condition | Setting Condition |
| TSG3nO1, TSG3nO3, TSG3nO5 | PWM output | (TSG3nCMP0E + 1) x count clock | Output an inactive level throughout one period (duty 0%) | TSG3nCMPmE = TSG3nCMP (m + 1) E or TSG3nCMP (m + 1) E > TSG3nCMP0E (m = 1, 5, 9) |
| | | | Output an active level of one count clock in one period | TSG3nCMPmE = TSG3nCMP (m + 1) E + 1 TSG3nCMP (m + 1) E = TSG3nCMPmE - 1 (m = 1, 5, 9) |
| | | | Output an inactive level of one count clock in one period | TSG3nCMPmE = TSG3nCMP (m + 1) E - 1 TSG3nCMP (m + 1) E = TSG3nCMPmE + 1 (m = 1, 5, 9) |
| | | | Output an active level throughout one period (duty 100%) | TSG3nCMPmE > TSG3nCMP0E TSG3nCMP (m + 1) E ≤ TSG3nCMP0E (m = 1, 5, 9) |
| TSG3nO2, TSG3nO4, TSG3nO6 | PWM output | (TSG3nCMP0E + 1) x count clock | Output an inactive level throughout one period (duty 0%) | TSG3nCMP (m - 1) E > TSG3nCMP0E (m = 2, 6, 10) |
| | | | Output an active level of one count clock in one period | TSG3nCMPmE = TSG3nCMP (m - 1) E - 1 TSG3nCMP (m - 1) E = TSG3nCMPmE + 1 (m = 2, 6, 10) |
| | | | Output an inactive level of one count clock in one period | TSG3nCMPmE = TSG3nCMP (m - 1) E + 1 TSG3nCMP (m - 1) E = TSG3nCMPmE - 1 (m = 2, 6, 10) |
| | | | Output an active level throughout one period (duty 100%) | TSG3nCMPmE = TSG3nCMP (m - 1) E or TSG3nCMPmE > TSG3nCMP0E (m = 2, 6, 10) |
| TSG3nO7 | Diagnostic signal output or pulse output by A/D conversion trigger | (TSG3nCMP0E + 1) x count clock | Please refer to Section 25.4.5, A/D Conversion Trigger Function. | |

(3) Various Settings of SP-PWM Mode

Setting mode

SP-PWM mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 010_B.

Setting timer output

The output pins TSG3nO1-TSG3nO6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.

The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

Error interrupt (INTTSG3nIER) due to the detection of the simultaneous active state of the positive and inverse phases is enabled by setting TSG3nIOC1.TSG3nEOC to 1. For details, refer to **Section 25.4.6, Error/Warning Interrupt**.

Setting rewriting timing of register with reload function

With the TSG3nCTL3.TSG3nRMC, reload (simultaneous rewrite) or rewrite (anytime) is specified for the registers with reload function. The default setting is 0 (reload). To reload, set TSG3nCTL4.TSG3nPRE to 1.

No reload timing is generated when TSG3nPRE = 0.

When “anytime rewrite” is specified, unintended output may be generated depending on the rewrite timing.

Setting A/D conversion trigger output

The A/D conversion trigger 0 (TSG3nADTRG0 signal) is set with TSG3nCTL5.TSG3nAT09 to TSG3nAT00.

TSG3nAT09 to TSG3nAT00 is used to enable or disable the A/D conversion trigger output at the match of TSG3nDCMP2E to TSG3nDCMP0E with the 18-bit counter (up count).

TSG3nCTL6.TSG3nAT19 to TSG3nAT10 is used to set the A/D conversion trigger 1 (TSG3nADTRG1 signal).

To set the match timing of the 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E, set the compare value to each register.

The skipping function can be used for TSG3nADTRG0 and TSG3nADTRG1 signals. TSG3nACC00 and TSG3nACC01 of TSG3nCTL5, and TSG3nACC10 and TSG3nACC11 of TSG3nCTL6 can be used to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

- Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E to TSG3nDCMP0E correctly when the timing pulse of A/D conversion trigger is output to TSG3nO7.
- In SP-PWM mode, no valley interrupt (INTTSG3nIVLY) is generated. Therefore, TSG3nCTL5.TSG3nAT00 and TSG3nCTL6.TSG3nAT10 must be set to 0.
- In SP-PWM mode, the 18-bit sub-counter does not operate. Therefore, TSG3nAT09 and TSG3nAT08 in TSG3nCTL5, and TSG3nAT19 and TSG3nAT18 in TSG3nCTL6 must be set to 0.

- In SP-PWM mode, down counting by the 18-bit counter is not generated. Therefore, TSG3nAT07, TSG3nAT05, and TSG3nAT03 in TSG3nCTL5 and TSG3nAT17, TSG3nAT15, and TSG3nAT13 in TSG3nCTL6 should be set to 0.

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions:

$$\text{PCLK} \times \text{TSG3nDTC0}$$

$$\text{PCLK} \times \text{TSG3nDTC1}$$

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the inactive state, and a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

Carrier period

Set the carrier period with TSG3nCMP0E according to the following expression:

$$\text{TSG3nCMP0E} = (\text{carrier period/count clock period}) - 1$$

CAUTION

PWM output with 100% duty cannot be produced when $\text{TSG3nCMP0E} = 3\text{FFFF}_{\text{H}}$.

Setting duty (PWM width)

The duty of U phase, V phase, and W phase is set with TSG3nCMPmE, TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE (m = 1, 2, 5, 6, 9, 10), respectively.

- The set timings of the U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E, and the clear timings are configured using TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E (when set timing and clear timing are used for control).
- The set timings of U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E while the active periods of these phases are configured using TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE.
The sum of the set values of TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E, and the set values of TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE are set to TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E respectively (when set timing and active period are used for control).

(4) Dead Time Control in SP-PWM mode

Duty setting registers are TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10), TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE and register for setting the period is TSG3nCMP0E. The 6-phase PWM waveform of a variable duty is output by using these registers. To achieve the dead time control, there are six 10-bit down counters that operate synchronously with the count clock of the 18-bit counter and two dead time control registers (TSG3nDTC0W and TSG3nDTC1W). TSG3nDTC0W is used for setting a dead time from a change of the inverse phase to the inactive state to a change of the positive phase to the active state. TSG3nDTC1W is used for setting a dead time from a change of the positive phase to the inactive state to a change of the inverse phase to the active state.

Dead time counter keeps operating even when operation stop (TSG3nTE = 0) setting collided with dead time insert timing, and the dead time set in TSG3nO1 and TSG3nO2 are always inserted.

The following figure shows an example of the output waveform.

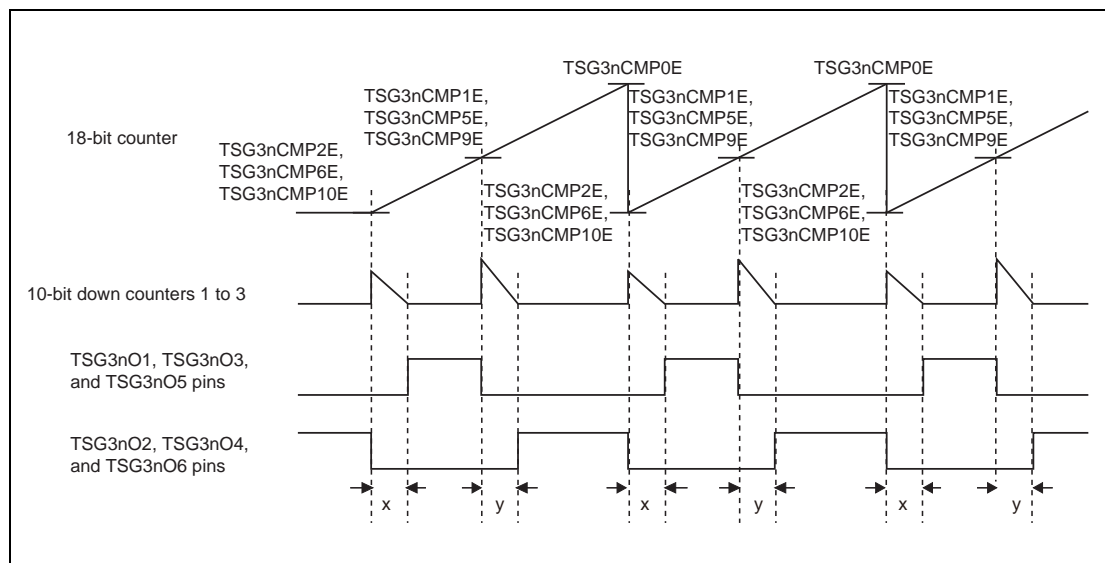


Figure 25.73 Example of Output Waveform in SP-PWM Mode

NOTE

x: TSG3nDTC0 register values, y: TSG3nDTC1 register values

(5) Software Output Control Function in SP-PWM Mode

TSG3nOPT0.TSG3nSOC, TSG3nIDC, and TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 are used to control timer output by software.

As shown in **Figure 25.74**, the output control is switched immediately when TSG3nSOC is set to 1. If the dead time is set, the period of the dead time is guaranteed. After that, when TSG3nSOC is set to 0, output control is retained, and at the reload timing, output control is switched to SP-PWM mode output control.

For details, refer to **Section 25.4.7.8, Software Output Control Function**.

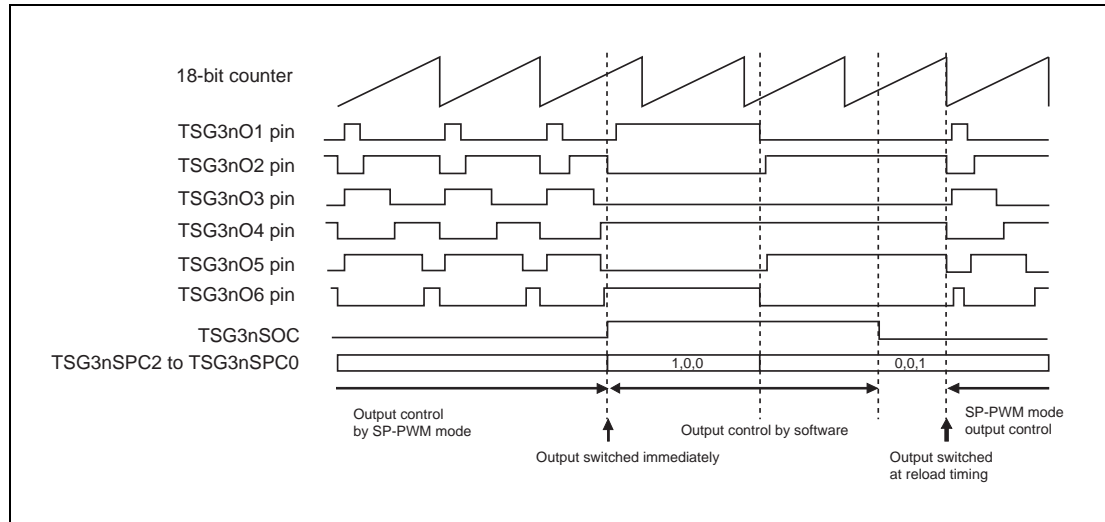


Figure 25.74 Example of Switching from SP-PWM Mode Control to Software Output Control

(a) Procedure for Software Output Control Processing

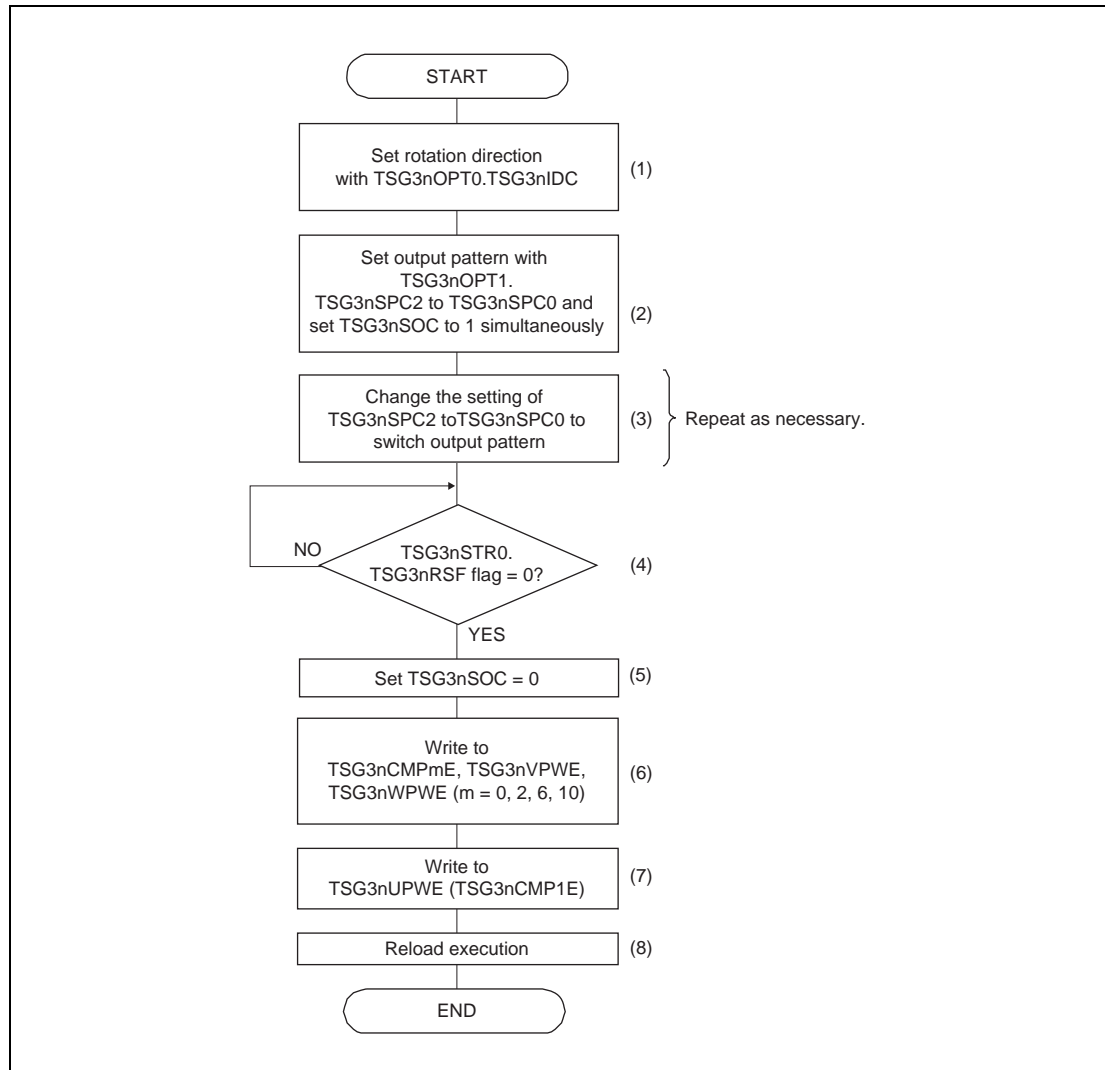


Figure 25.75 Flow of Software Output Control

The procedure for software output control is described below.

- (1) Set TSG3nIDC to determine the electric current direction. The timer output has a 180-degree phase shift between when TSG3nIDC = 0 and when TSG3nIDC = 1. When this bit is rewritten with the software output control function, the output pattern will change according to the new setting at the next timer cycle.
- (2) Set the output pattern in TSG3nSPC2-0. At the same time, set TSG3nSOC to 1 to switch to software output control mode.
- (3) Change the output pattern setting for TSG3nSPC2-0 to change the timer output.
- (4) Ensure that the reload request flag TSG3nRSF is 0. If TSG3nRSF is 1, do not proceed to the following step until TSG3nRSF = 0.
- (5) By clearing TSG3nSOC to 0, software control release process starts (it is not released yet at this point).
- (6) Configure the compare register settings that are required after the software output control is released. Proceed to the following step when the register configuration is not required. When configuring the registers with the reload function, configure them at this point.
- (7) Write to TSG3nUPWE (TSG3nCMP1E) to start reloading.
- (8) Reload is executed and software output control is released.

CAUTION

Be sure to execute reload after completing steps (4), (5), (6), and (7). Unless reload can be executed, software output control cannot be released

25.4.7.4 120-DC Mode

Overview

In this mode, PWM output period set to TSG3nCMP0E and timer output (TSG3nO1 to TSG3nO6) according to the duty cycle set with TSG3nCMP1E to TSG3nCMP12E are controlled with three types of pattern inputs (software output control method, pattern switch method, and trigger switch method) to perform 120-DC control.

Prerequisites

- Set the PWM period with TSG3nCMP0E.
- Set the PWM duty with TSG3nCMP1E to TSG3nCMP12E and set the output pattern with TSG3nPAT0W and TSG3nPAT1W.

Functional description

In this mode, configure the PWM period, the duty cycle with each compare register, and the pattern to be output with each pattern register. Counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter starts counting from 00000_H, and is cleared by the match with TSG3nCMP0E.

INTTSG3nI1 to INTTSG3nI12 interrupts are generated by a compare match of the 18-bit counter and TSG3nCMP1E to TSG3nCMP12E buffer registers, respectively.

NOTE

120-DC mode is enabled when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 011_B.

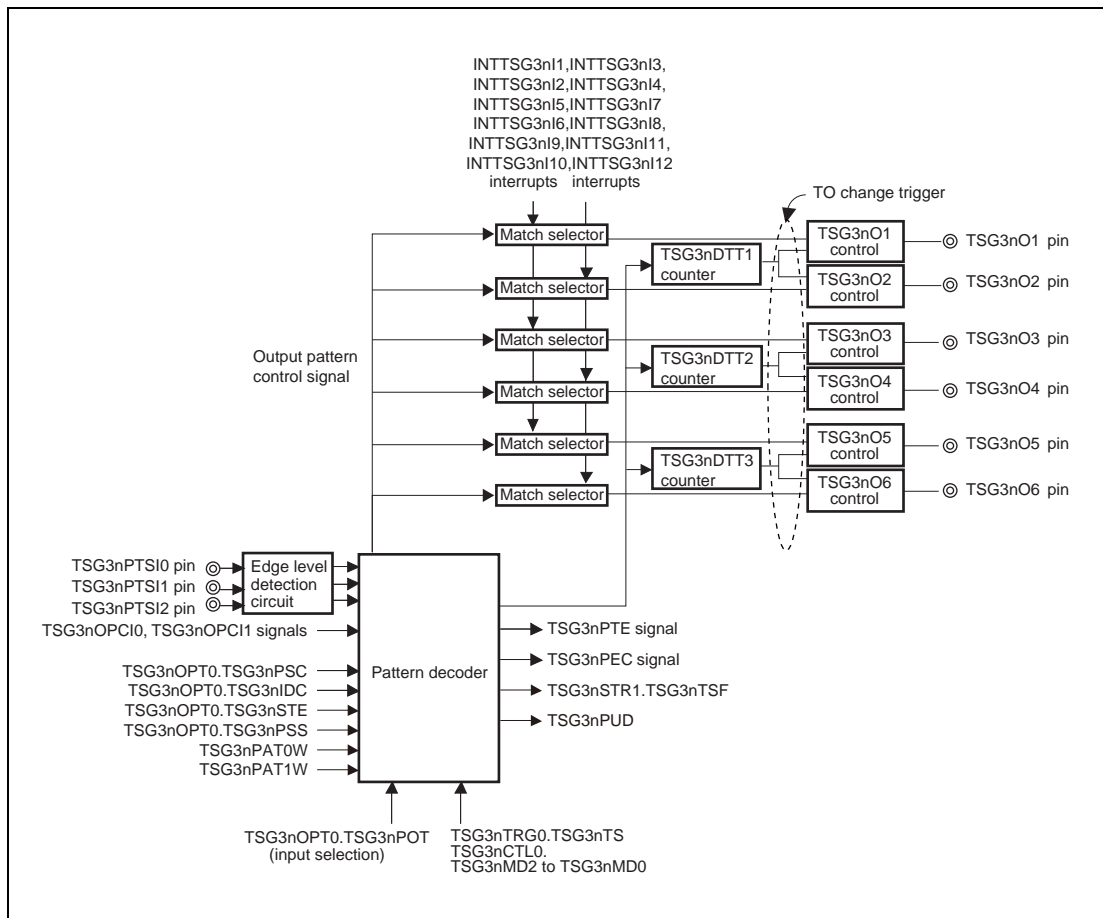


Figure 25.76 Block Diagram in 120-DC Mode

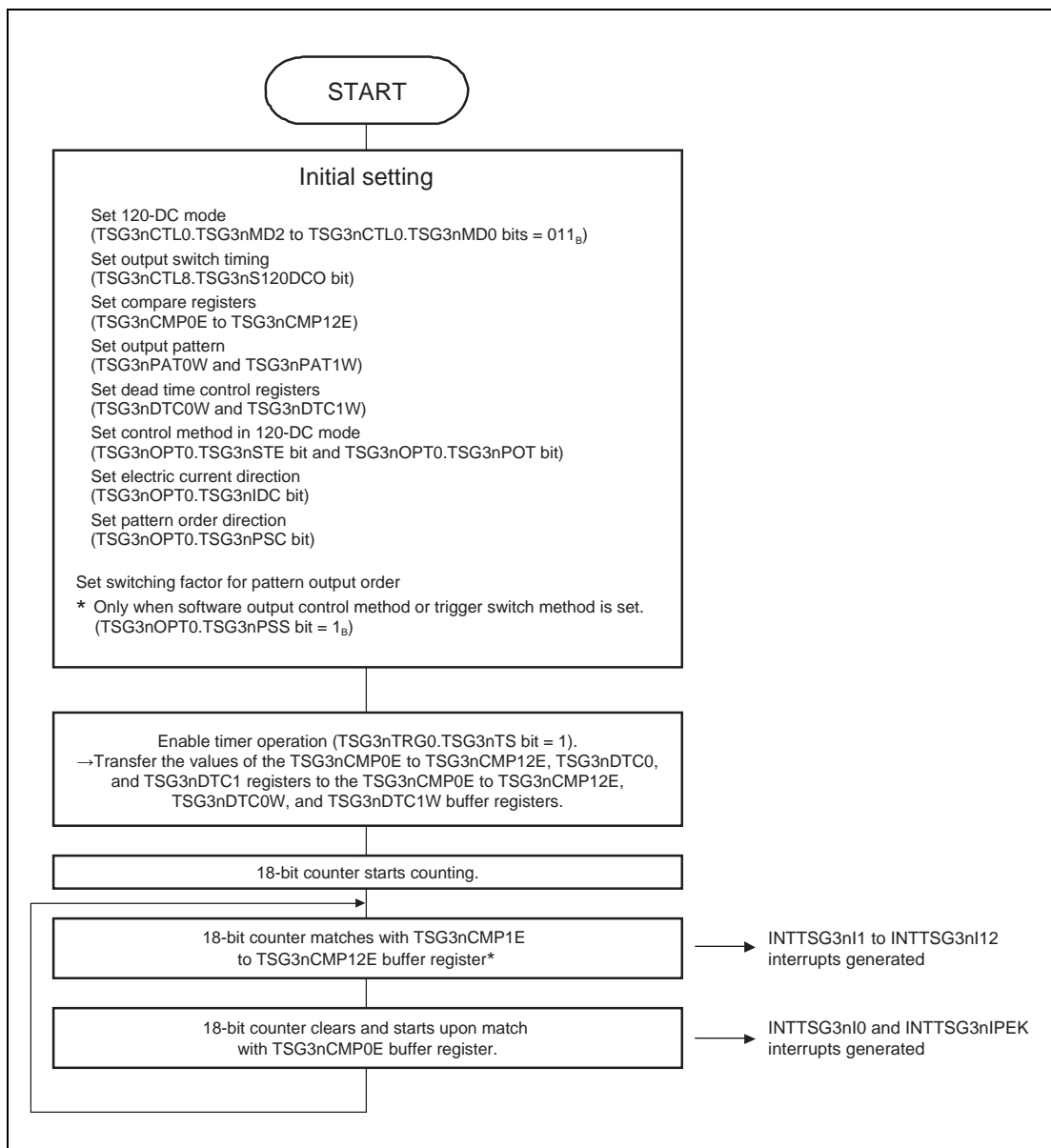


Figure 25.77 Basic Operation Flow in 120-DC Mode

NOTE

The 18-bit counter is not cleared by match of the 18-bit counter with the TSG3nCMP1E to TSG3nCMP12E buffer registers

(1) List of Operations in 120-DC Mode

Table 25.74 Counter Functions in 120-DC Mode

| Operation | Setting Condition | |
|----------------|-------------------|--|
| 18-bit counter | Start | TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger |
| | Clear | When TSG3nCTL8.TSG3nS120DCO is 0: timing of a match of TSG3nCMP0E value and 18-bit counter value or output pattern switch When TSG3nCTL8.TSG3nS120DCO is 1: timing of a match of TSG3nCMP0E value with 18-bit counter value |
| | Stop | TSG3nTRG1.TSG3nTT = 0 → 1 |

Table 25.75 Functions of Compare Registers and Dead Time Control Registers in 120-DC Mode

| Register | Rewrite Method | Rewrite during Operation | Function |
|---|----------------|--------------------------|--|
| TSG3nCMP0E | Reload | Possible | Setting period |
| TSG3nCMPmE (m = 1-12) | Reload | Possible | Setting PWM duty |
| TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E | Reload | Possible | Diagnostic signal output or A/D conversion trigger |
| TSG3nDTC0W, TSG3nDTC1W | Reload | Possible | Setting dead time |

Table 25.76 Timer Input Function in 120-DC Mode

| Pin/Signal | Function |
|--------------------------------|--------------------------|
| TSG3nCLKI pin | Clock enable input |
| TSG3nPTSI2-TSG3nPTSI0 pins | Pattern input (3 phases) |
| TSG3nOPCI0, TSG3nOPCI1 signals | Trigger input |

Table 25.77 Timer Output Function in 120-DC Mode

| Pin/Signal | Function |
|-----------------|--|
| TSG3nO1 pin | PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) with the 18-bit counter and output pattern selected by TSG3nPAT0W setting. |
| TSG3nO2 pin | PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 3, 4, 7, 8, 11, 12) with the 18-bit counter and output pattern selected by TSG3nPAT1W setting. |
| TSG3nO3 pin | PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) with the 18-bit counter and output pattern selected by TSG3nPAT0W setting. |
| TSG3nO4 pin | PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 3, 4, 7, 8, 11, 12) with the 18-bit counter and output pattern selected by TSG3nPAT1W setting. |
| TSG3nO5 pin | PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) with the 18-bit counter and output pattern selected by TSG3nPAT0W setting. |
| TSG3nO6 pin | PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 3, 4, 7, 8, 11, 12) with the 18-bit counter and output pattern selected by TSG3nPAT1W setting. |
| TSG3nO7 pin | Diagnostic signal output or pulse output by A/D conversion trigger |
| TSG3nPTE signal | Toggle signal by change in input pattern |

Table 25.78 Interrupt Requests in 120-DC Mode

| Interrupt | Function |
|--------------------------|--|
| INTTSG3nIm (m = 0 to 12) | Compare match of TSG3nCMPmE buffer register and 18-bit counter (m = 0 to 12) |
| INTTSG3nIER | Error |
| INTTSG3nIVLY | — |
| INTTSG3nIPEK | Peak interrupt (generated at the same timing as INTTSG3nI0) |
| INTTSG3nIWN | Warning interrupt |

Table 25.79 Compare Match Timing in 120-DC Mode

| Compare Match | Timing |
|--------------------------|--|
| TSG3nCMP0E | When 18-bit counter changes from TSG3nCMP0E to 00000 _H |
| TSG3nCMPmE (m = 1 to 12) | After detecting the match of 18-bit counter and TSG3nCMPmE (m = 1 to 12) |

Table 25.80 Example of Setting Each Timer Output Condition in 120-DC Mode

| Pin | Item | Output Period | Output Duty | |
|-------------------------|---|---|--|-------------------|
| | | | Output Condition | Setting Condition |
| TSG3nOm (m = 1 to 6) | PWM output | $(TSG3nCMP0E + 1) \times \text{count}$ clock | See (6), List of Output Patterns in 120-DC Mode. | — |
| TSG3nO7 | Diagnostic signal output or pulse output by A/D conversion trigger | $(TSG3nCMP0E + 1) \times \text{count}$ clock | See Section 25.4.5, A/D Conversion Trigger Function. | — |

(2) Various Settings of 120-DC Mode

Setting mode

120-DC mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 011_B.

Setting timer output

The output pins TSG3nO1 to TSG3nO6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.

The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

With TSG3nIOC1.TSG3nEOC = 1, the error interrupt (INTTSG3nIER) generation is enabled when the simultaneous active state of the positive phase and inverse phase is detected. For details, see **Section 25.4.6, Error/Warning Interrupt**.

Setting register rewrite timing

Reloading the registers with the reload function is activated with TSG3nCTL3.TSG3nRMC (simultaneous rewrite; default setting is 0 = reload). To reload, set TSG3nCTL4.TSG3nPRE to 1.

The reload timing is not generated if TSG3nPRE is 0.

Setting A/D conversion trigger output

To set A/D conversion trigger 0 (TSG3nADTRG0 signal), use TSG3nCTL5.TSG3nAT09 to TSG3nAT00.

With TSG3nAT09 to TSG3nAT00, A/D conversion trigger output is enabled or disabled at the match of the 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E (during up count).

To set A/D conversion trigger 1 (TSG3nADTRG1 signal), use TSG3nCTL6.TSG3nAT19 to TSG3nAT10.

To set the match timing of the 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E, set the compare value to the pertinent register.

The skipping function can be used for TSG3nADTRG0 and TSG3nADTRG1 signals. Use TSG3nACC01, TSG3nACC00 of TSG3nCTL5, and TSG3nACC11, TSG3nACC10 of TSG3nCTL6 to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

- Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E to TSG3nDCMP0E correctly when the timing pulse of A/D conversion trigger is output to TSG3nO7.
- In 120-DC mode, a valley interrupt (INTTSG3nIVLY) is not generated. Therefore, TSG3nCTL5.TSG3nAT00 and TSG3nCTL6.TSG3nAT10 must be set to 0.
- In 120-DC mode, the 18-bit sub-counter does not operate. TSG3nAT09 and TSG3nAT08 in TSG3nCTL5, and TSG3nAT19 and TSG3nAT18 in TSG3nCTL6 must be set to 0.
- In 120-DC mode, the 18-bit counter does not decrement. Therefore, TSG3nAT07, TSG3nAT05, and TSG3nAT03 in TSG3nCTL5 and TSG3nAT17, TSG3nAT15, and

TSG3nAT13 in TSG3nCTL6 must be set to 0.

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions

$$\text{PCLK} \times \text{TSG3nDTC0}$$

$$\text{PCLK} \times \text{TSG3nDTC1}$$

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the inactive state to a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

Carrier period

Set the carrier period with TSG3nCMP0E according to the following expression:

$$\text{TSG3nCMP0E} = (\text{carrier period/count clock period}) - 1$$

Setting duty (PWM width)

The duty of PWM output is set with TSG3nCMP1E to TSG3nCMP12E. The setting range of the compare registers is as follows:

$$00000_{\text{H}} \leq \text{TSG3nCmPmE} \leq \text{TSG3nCMP0E} + 1$$

CAUTION

Do not set TSG3nCmPmE = TSG3nCMP0E + 1 (m = 1 to 12) only when TSG3nCMP0E + 1 < TSG3nCmPmE, and TSG3nCMP0E = 3FFFF_H.

Setting output PWM

In 120-DC mode, the output pins TSG3nO1, TSG3nO3, and TSG3nO5 are controlled by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E, and the output pins TSG3nO2, TSG3nO4, TSG3nO6 are controlled by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E. The duty cycle of a PWM period (TSG3nCMP0E) can be set with TSG3nCMP1E to TSG3nCMP12E. Setting TSG3nCMP1E to TSG3nCMP12E to 00000_H sets the PWM duty cycle to 0%. Setting TSG3nCMP1E to TSG3nCMP12E to TSG3nCMP0E + 1 value sets the PWM duty cycle to 100%. This allows chopping output control and rectangular wave output control.

(3) Control Methods in 120-DC Mode

Control methods in 120-DC mode are listed below.

Table 25.81 Control Method in 120-DC Mode

| Control Method | Function |
|--------------------------------|---|
| Software output control method | Switches the output pattern according to the TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 setting made by software. |
| Pattern switch method | Switches the output pattern by the pattern input signal of TSG3nPTSI0 to TSG3nPTSI2. |
| Trigger switch method | Switches the output pattern by the trigger switch method using the trigger input signals TSG3nOPCI0 and TSG3nOPCI1 or by the pattern input setting of TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 in the constant order. |

Switch timing of timer output when changing input pattern of 120-DC mode can be set by TSG3nCTL8.TSG3nS120DCO.

Table 25.82 Setting of TSG3nS120DCO and Operation in 120-DC Mode

| TSG3nS120DCO | Function |
|--------------|---|
| 0 | When input patterns are changed, the main counter (TSG3nCNTE) is cleared and the change of patterns is immediately reflected to timer output. |
| 1 | When input patterns are changed, the change of input patterns is reflected to timer output at the next timer cycle (after a match of the main counter (TSG3nCNTE) with TSG3nCMP0E). |

Setting software output control method

Setting TSG3nOPT0.TSG3nSTE = 0 enables switching of output pattern by software output control. The TSG3nO1 to TSG3nO6 pin output is switched according to the settings of TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0.

The pattern output order at the beginning of operation is set with TSG3nOPT0.TSG3nIDC and TSG3nOPT0.TSG3nPSC.

Operation of software output control method

The PWM output of TSG3nO1 to TSG3nO6 pins (PWM output defined by TSG3nCMP1E to TSG3nCMP1E) is selected using TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 configured by software. As for the control of the dead time, the dead time counter is activated at the falling edge of the signals in each phase and the dead time is inserted.

The 18-bit counter counts according to the carrier period set in TSG3nCMP0E. The 18-bit counter is cleared by match of the 18-bit counter and TSG3nCMP0E or by a write access to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 (when TSG3nS120DCO = 0).

In this method, the pattern, which is decoded using information consisting of the output pattern (TSG3nSPC2 to TSG3nSPC0), the electric current direction control bit (TSG3nOPT0.TSG3nIDC), and order detection control bit (TSG3nOPT0.TSG3nPSC), is output. **Figure 25.98** shows the timer output when the output pattern is changed by software output control.

Immediately after the operation starts (TSG3nTRG0.TSG3nTS = 1), the output pattern of TSG3nSPC2 to TSG3nSPC0 and the pattern set with TSG3nIDC and TSG3nPSC (TSG3nOPT0.TSG3nPSS = 1) are output.

Setting pattern switch method

Setting TSG3nOPT0.TSG3nSTE to 1 and TSG3nPOT to 0 selects the pattern switch method. The TSG3nO1 to TSG3nO6 pin output pattern is changed at the change timing of the TSG3nPTS12 to TSG3nPTS10 pins.

The output pattern at the beginning of operation is set with TSG3nOPT0.TSG3nIDC and with TSG3nOPT0.TSG3nPSC. However, after determining the rotation direction (after the value is set to TSG3nSTR1.TSG3nTSF), the setting of TSG3nPSC is disabled.

Operation of pattern switch method

After level detection is performed for the pins (three inputs from the hall sensor), the level-detected signals are decoded. From the decoding result, the PWM output of TSG3nO1 to TSG3nO6 pins (PWM output defined by TSG3nCMP1E to TSG3nCMP12E value) is selected. To control the dead time, the dead time counter is activated at the falling timing of signals in each phase and the dead time is inserted.

The 18-bit counter is cleared by a match of the 18-bit counter value with the TSG3nCMP0E value or by a change of the input pattern (TSG3nPTS12 to TSG3nPTS10 pins) while TSG3nS120DCO is 0.

In this method, the pattern, which is decoded by using information on input pattern (TSG3nPTS12 to TSG3nPTS10), the electric current direction control bit (TSG3nOPT0.TSG3nIDC), and TSG3nPTS12 to TSG3nPTS10 pattern order detection flag (TSG3nSTR1.TSG3nTSF), is output. **Figure 25.80** to **Figure 25.83** show the timer output when TSG3nPTS12 to TSG3nPTS10 pin inputs change.

Immediately after the operation starts (TSG3nTRG0.TSG3nTS = 1), the output pattern set by the levels input on the TSG3nPTS12 to TSG3nPTS10 pins and by the TSG3nIDC and TSG3nPSC bits is produced. If a level on any of the TSG3nPTS12 to TSG3nPTS10 pins is changed, TSG3nTSF is determined by the direction of the change to the order. After the TSG3nTSF value is determined, the pattern set by the TSG3nTSF bit replaces the pattern set by the TSG3nPSC bit.

CAUTION

When connecting the three-phase pulse input signal to the TSG3nPTS12 to TSG3nPTS10 pins, confirm that the three-phase pulse input value and the patterns output from the TSG3nO1 to TSG3nO6 pins satisfy the expected conditions. If the expected conditions are not satisfied, change the connection between the three-phase pulse input signal and the TSG3nPTS12 to TSG3nPTS10 pins.

Setting trigger switch method

Setting TSG3nOPT0.TSG3nSTE and TSG3nPOT to 1 selects the trigger switch method. The output patterns of pins TSG3nO1 to TSG3nO6 are changed at a rising edge of an external input (TSG3nOPCI1 and TSG3nOPCI0 signals).

For pattern output order, see **Section 25.4.7.4, (5) Operation in 120-DC Mode**.

The initial output pattern can be controlled with TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0.

The initial pattern is output when the TSG3n operation starts (TSG3nTRG0.TSG3nTS = 1) after TSG3nSPC2 to TSG3nSPC0 are configured. For details, see **Section 25.4.7.4, (6) List of Output Patterns in 120-DC Mode**.

Operation of trigger switch method

With the trigger input switch method, the rising edges of the TSG3nOPCI0 and TSG3nOPCI1 signals are detected and the output switch timing is generated. The initial timer output pattern is set with TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0. The subsequent output patterns are switched after rising of the TSG3nOPCI0 and TSG3nOPCI1 signals is detected. Furthermore, the output patterns can be switched by setting TSG3nOPT1.TSG3nSPC2 to TSG3nOPT1.TSG3nSPC0.

The 18-bit counter counts based on the carrier period set in TSG3nCMP0E. The 18-bit counter is cleared by match of the 18-bit counter and TSG3nCMP0E, by a write access to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0, or by detecting a rising of TSG3nOPCI0 and TSG3nOPCI1 signals. (When TSG3nS120DCO = 0)

For examples of operation in 120-DC mode when trigger input switch method is used, see **Figure 25.80** to **Figure 25.83**.

CAUTION

The initial pattern should be set according to the read input level of the port to which TSG3nPTS12 to TSG3nPTS10 pins are connected.

(4) Timer Output in 120-DC Mode

In 120-DC mode, the PWM output is controlled with TSG3nPAT0W, TSG3nPAT1W, and TSG3nCMP1E to TSG3nCMP12E. TSG3nPAT0W, TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E are set to control the output of TSG3nO1, TSG3nO3, and TSG3nO5 pins. TSG3nPAT1W, TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E are set to control the output of TSG3nO2, TSG3nO4, and TSG3nO6 pins.

With PWM output control, eight types of output patterns can be selected for each of TSG3nO1, TSG3nO3, and TSG3nO5 pins and TSG3nO2, TSG3nO4, and TSG3nO6 pins.

Table 25.83 TSG3nPAT0W Set Value and Output Control

| PATmT Value | Output Control |
|-------------|---------------------------------|
| 000 | Fixed to low |
| 001 | PWM output set with TSG3nCMP1E |
| 010 | PWM output set with TSG3nCMP2E |
| 011 | PWM output set with TSG3nCMP5E |
| 100 | PWM output set with TSG3nCMP6E |
| 101 | PWM output set with TSG3nCMP9E |
| 110 | PWM output set with TSG3nCMP10E |
| 111 | Fixed to high |

Note: m = 0, 1, 2, 3, 4, 5

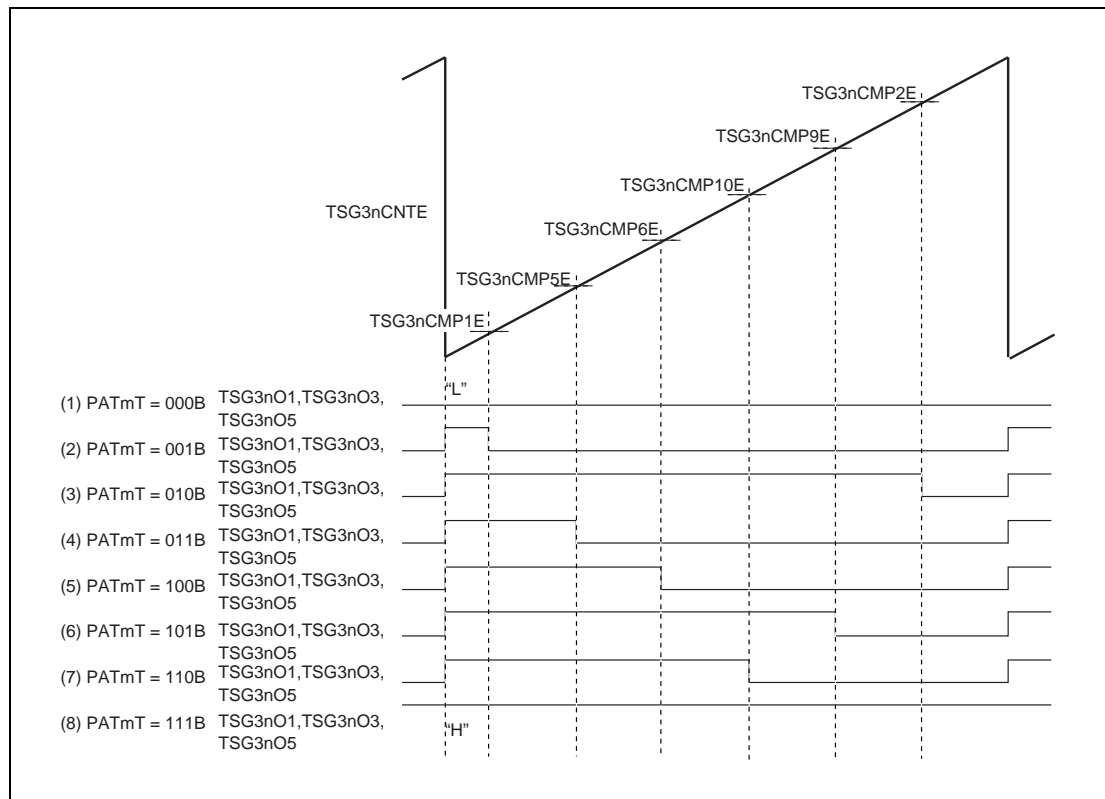


Figure 25.78 TSG3nO1, TSG3nO3, TSG3nO5 Pin Output of Each Output Pattern

Table 25.84 TSG3nPAT1W Set Value and Output Control

| PATmB Value | Output Control |
|-------------|---------------------------------|
| 000 | Fixed to low |
| 001 | PWM output set with TSG3nCMP3E |
| 010 | PWM output set with TSG3nCMP4E |
| 011 | PWM output set with TSG3nCMP7E |
| 100 | PWM output set with TSG3nCMP8E |
| 101 | PWM output set with TSG3nCMP11E |
| 110 | PWM output set with TSG3nCMP12E |
| 111 | Fixed to high |

Note: m = 0, 1, 2, 3, 4, 5

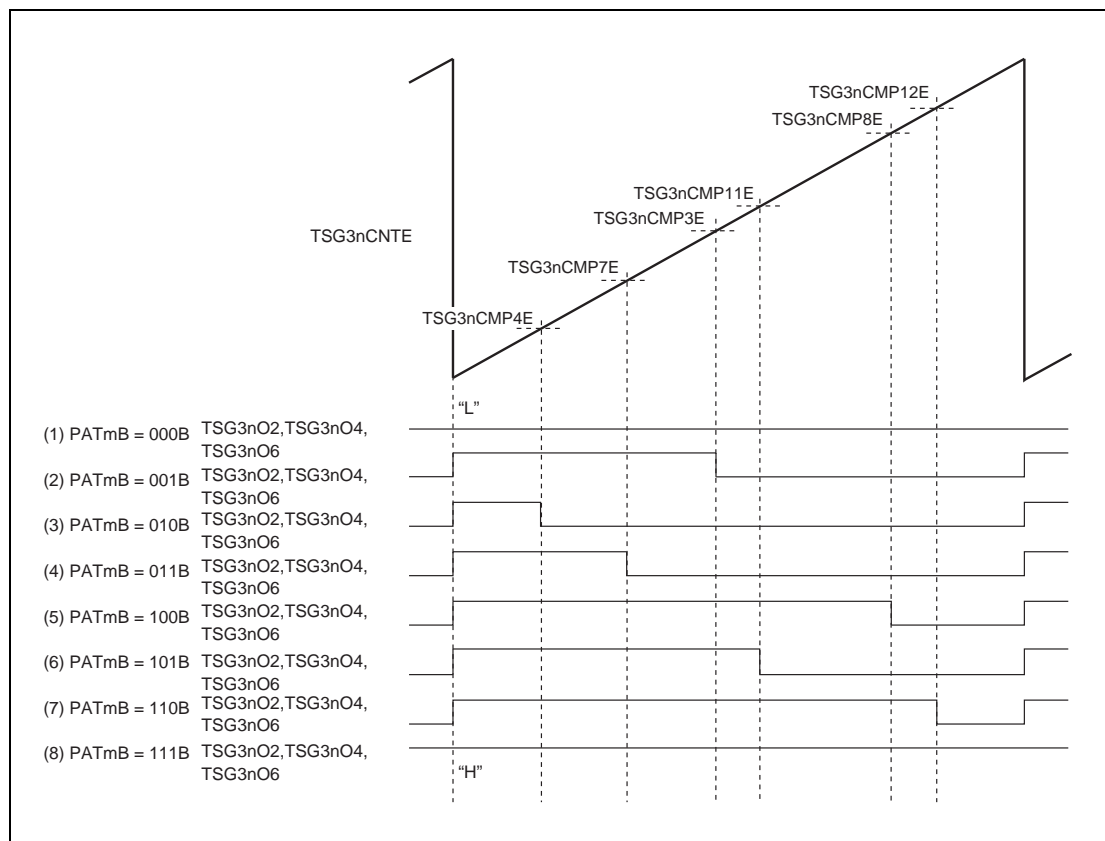


Figure 25.79 TSG3nO2, TSG3nO4, TSG3nO6 Pin Output of Each Output Pattern

(5) Operation in 120-DC Mode

Figure 25.80 to **Figure 25.83** show examples of operation in 120-DC mode.

The TSG3nO1 to TSG3nO6 pins detect the input level change of the TSG3nPTSI2 to TSG3nPTSI0 pins, and then change the output pattern. The 18-bit counter produces sawtooth waveform and PWM output using TSG3nCMP0E to TSG3nCMP12E.

When TSG3nS120DCO = 0, the 18-bit counter is cleared to 00000_H each time the counter value matches with TSG3nCMP0E or a change in the TSG3nPTSI2 to TSG3nPTSI0 pins is detected. The timer output pattern is switched each time a change in the TSG3nPTSI2 to TSG3nPTSI0 pins is detected.

When TSG3nS120DCO = 1, the 18-bit counter is cleared to 00000_H when the counter value matches with TSG3nCMP0E but not cleared with a change in the TSG3nPTSI2 to TSG3nPTSI0 pins. The timer output pattern is switched to the one corresponding to the patterns of TSG3nPTSI2 to TSG3nPTSI0 pins at the next match timing of TSG3nCNTE and TSG3nCMP0E.

NOTE

PAT0T to PAT5T and PAT0B to PAT5B show PWM operation set by TSG3nCMP1E to TSG3nCMP12E, respectively.

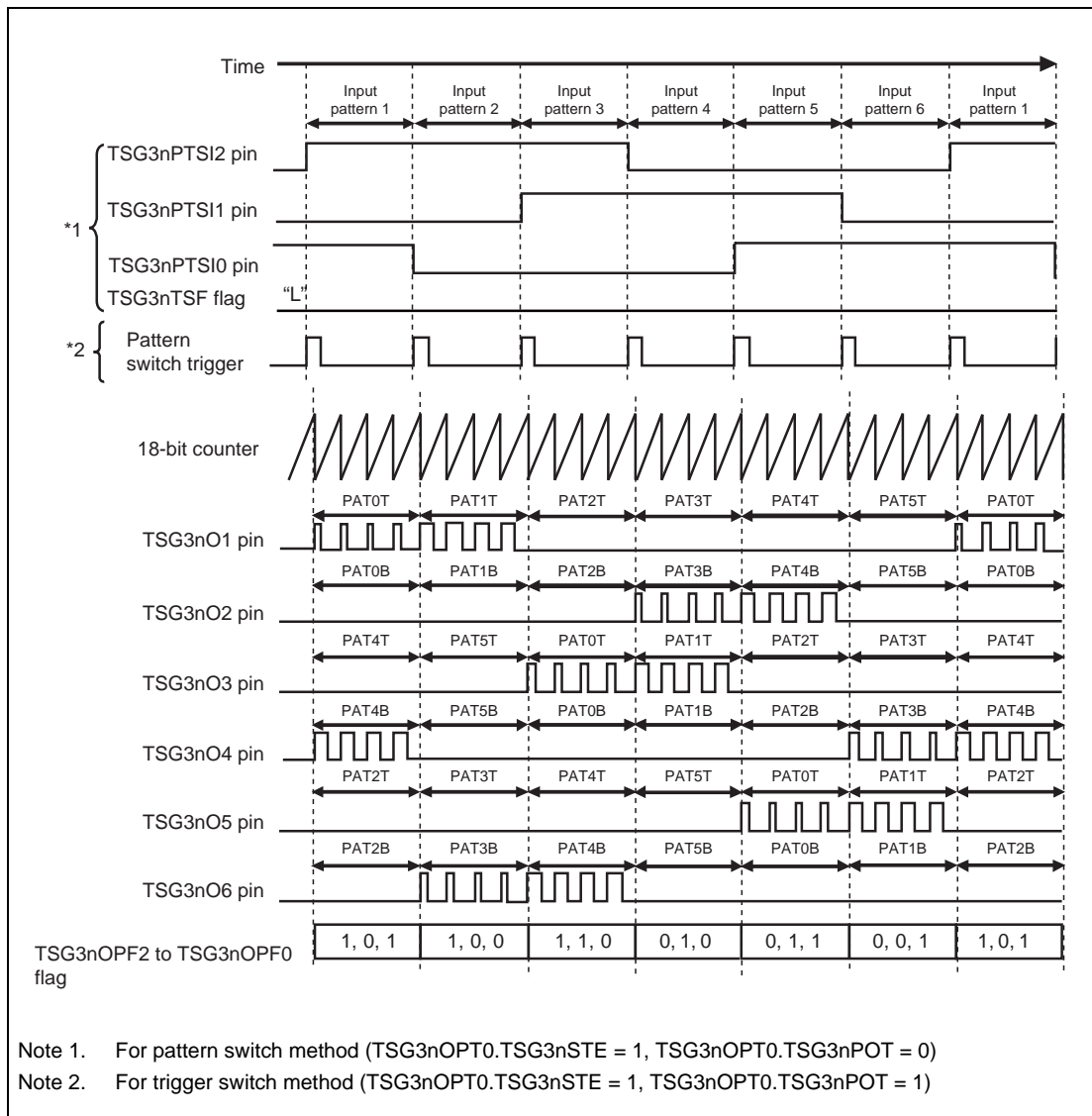


Figure 25.80 Example of Operation in 120-DC Mode (Normal Rotation: TSG3nSTR1.TSG3nTSF = 0 and TSG3nOPT0.TSG3nIDC = 0)

NOTE

TSG3nOPT0.TSG3nSOC = 0

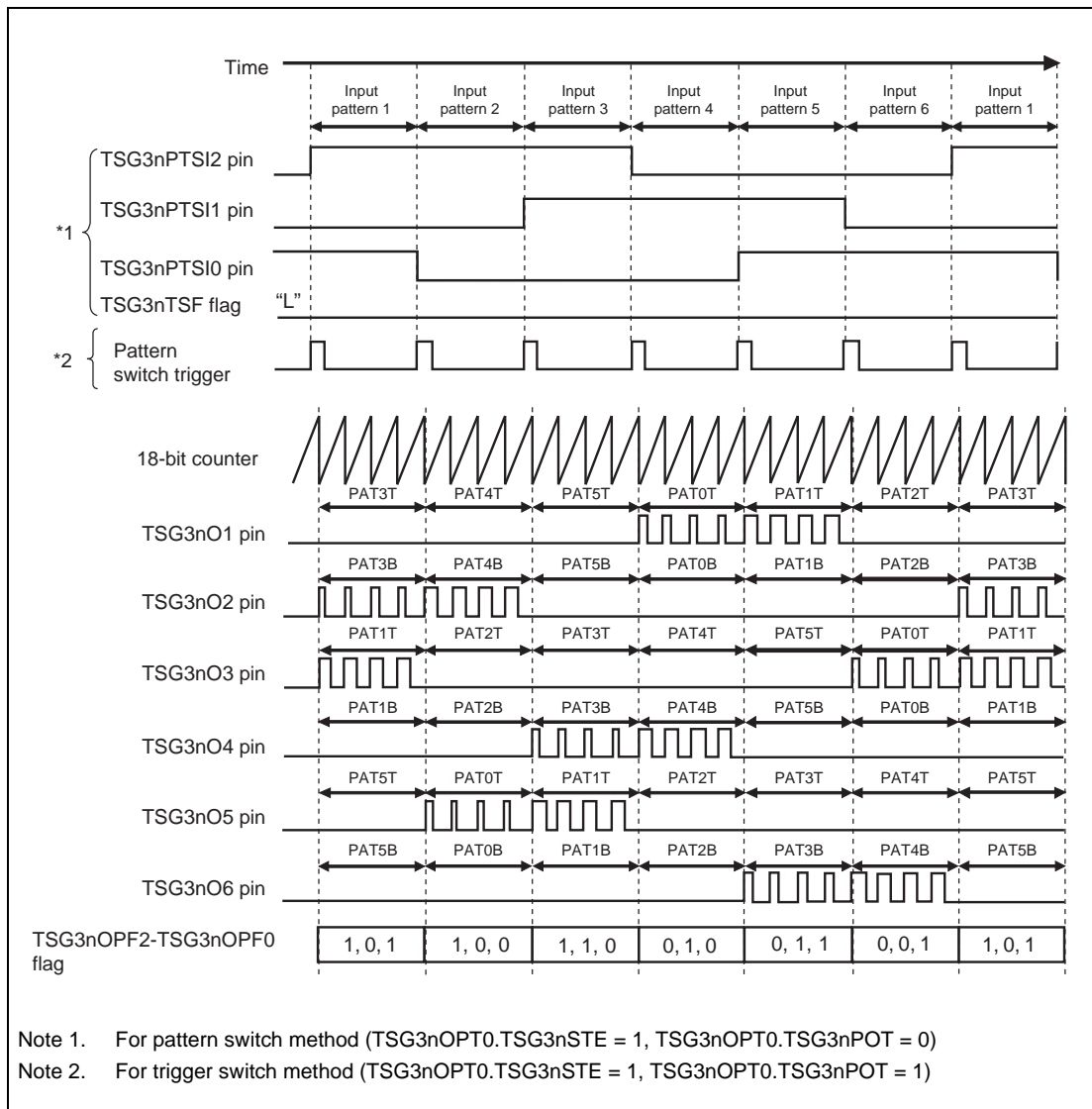


Figure 25.81 Example of Operation in 120-DC Mode (Normal Rotation: TSG3nSTR1.TSG3nTSF = 0 and TSG3nOPT0.TSG3nIDC = 1)

NOTE

TSG3nOPT0.TSG3nSOC = 0

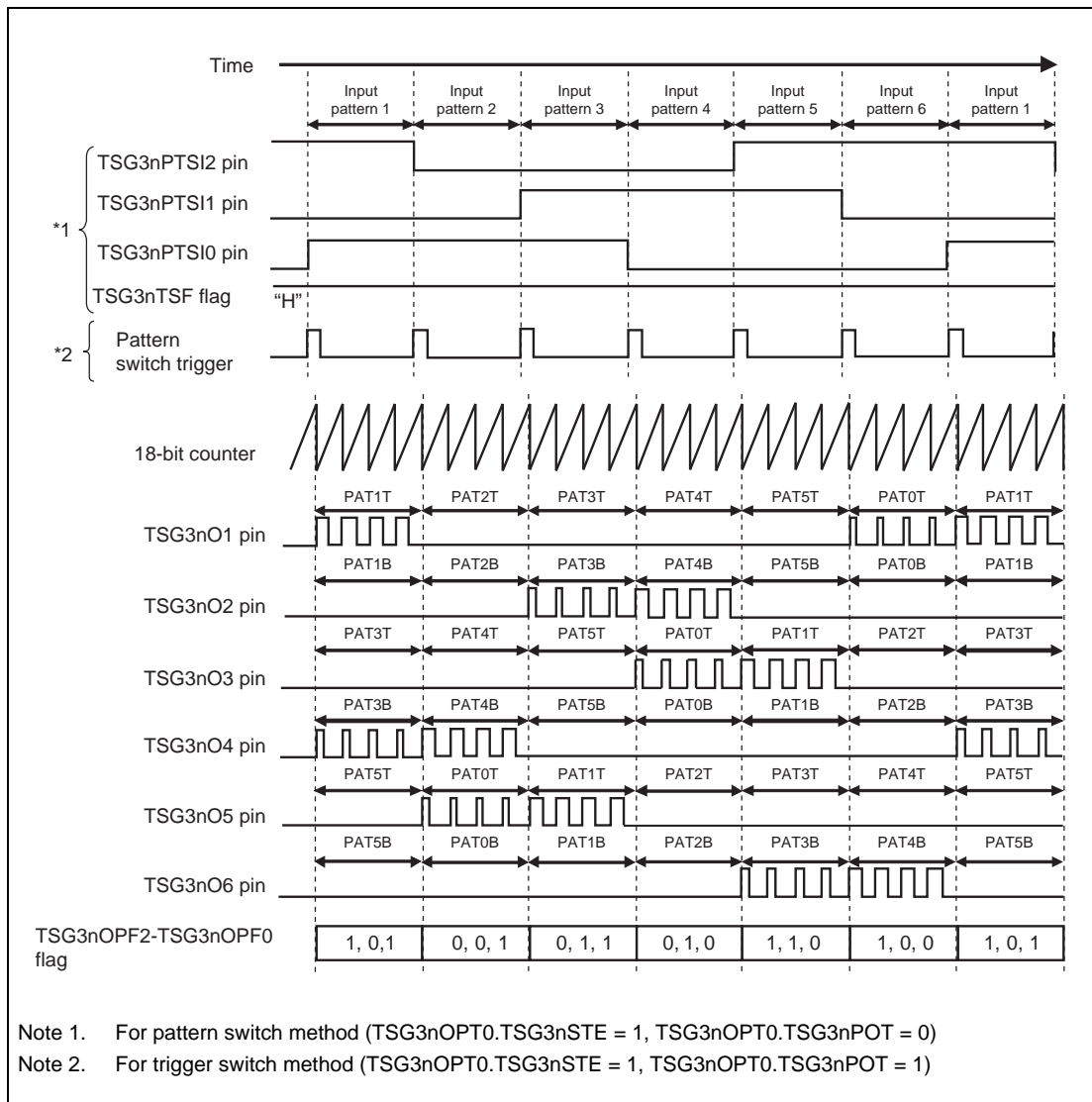


Figure 25.82 Example of Operation in 120-DC Mode (Reverse Rotation: TSG3nSTR1.TSG3nTSF = 1 and TSG3nOPT0.TSG3nIDC = 0)

NOTE

TSG3nOPT0.TSG3nSOC = 0

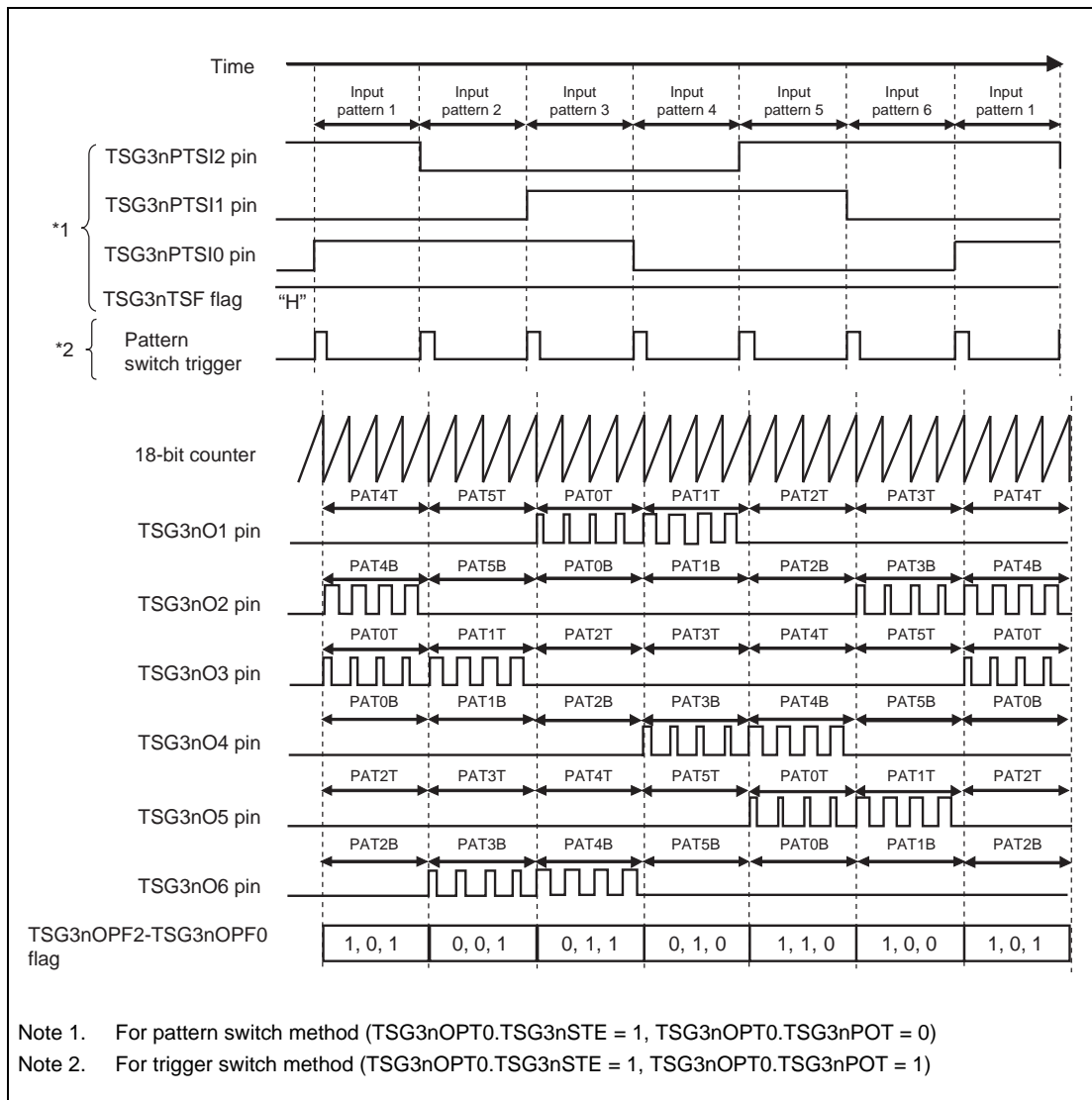


Figure 25.83 Example of Operation in 120-DC Mode (Reverse Rotation: TSG3nSTR1.TSG3nTSF = 1 and TSG3nOPT0.TSG3nIDC = 1)

NOTE

TSG3nOPT0.TSG3nSOC = 0

(6) List of Output Patterns in 120-DC Mode

In 120-DC mode, the output pattern is determined according to the electric current direction (TSG3nOPT0.TSG3nIDC) and the pattern order direction.

Table 25.85 Selection of Pattern Order Direction in 120-DC Mode

| TSG3nOPT0 | | | Pattern Order Direction |
|-----------|----------|----------|---|
| TSG3nSTE | TSG3nPOT | TSG3nPSS | |
| 0 | — | 1 | TSG3nPSC |
| 1 | 0 | — | TSG3nTSF (TSGnPSC only for initial setting at operation start) |
| 1 | 1 | 1 | TSG3nPSC |

Table 25.86 List of Output Patterns in 120-DC Mode (1/4)

Electric current direction: normal (TSG3nIDC = 0)

Pattern order direction: normal (TSG3nTSF = 0 or TSG3nPSC = 0)

| Output pins | TSG3nOPT1.TSG3nSPC2-TSG3nSPC0*1/TSG3nSTR1.TSG3nOPF2-TSG3nOPF0 | | | | | | | |
|-------------|---|-------|-------|-------|-------|-------|-----|-----|
| | 101 | 100 | 110 | 010 | 011 | 001 | 000 | 111 |
| TSG3nO1 | PAT0T | PAT1T | PAT2T | PAT3T | PAT4T | PAT5T | Low | Low |
| TSG3nO2 | PAT0B | PAT1B | PAT2B | PAT3B | PAT4B | PAT5B | Low | Low |
| TSG3nO3 | PAT4T | PAT5T | PAT0T | PAT1T | PAT2T | PAT3T | Low | Low |
| TSG3nO4 | PAT4B | PAT5B | PAT0B | PAT1B | PAT2B | PAT3B | Low | Low |
| TSG3nO5 | PAT2T | PAT3T | PAT4T | PAT5T | PAT0T | PAT1T | Low | Low |
| TSG3nO6 | PAT2B | PAT3B | PAT4B | PAT5B | PAT0B | PAT1B | Low | Low |

Table 25.86 List of Output Patterns in 120-DC Mode (2/4)

Electric current direction: reverse (TSG3nIDC = 1)

Pattern order direction: normal (TSG3nTSF = 0 or TSG3nPSC = 0)

| Output pins | TSG3nOPT1.TSG3nSPC2-TSG3nSPC0*1/TSG3nSTR1.TSG3nOPF2-TSG3nOPF0 | | | | | | | |
|-------------|---|-------|-------|-------|-------|-------|-----|-----|
| | 101 | 100 | 110 | 010 | 011 | 001 | 000 | 111 |
| TSG3nO1 | PAT3T | PAT4T | PAT5T | PAT0T | PAT1T | PAT2T | Low | Low |
| TSG3nO2 | PAT3B | PAT4B | PAT5B | PAT0B | PAT1B | PAT2B | Low | Low |
| TSG3nO3 | PAT1T | PAT2T | PAT3T | PAT4T | PAT5T | PAT0T | Low | Low |
| TSG3nO4 | PAT1B | PAT2B | PAT3B | PAT4B | PAT5B | PAT0B | Low | Low |
| TSG3nO5 | PAT5T | PAT0T | PAT1T | PAT2T | PAT3T | PAT4T | Low | Low |
| TSG3nO6 | PAT5B | PAT0B | PAT1B | PAT2B | PAT3B | PAT4B | Low | Low |

Note 1. When TSG3nSPC2 to TSG3nSPC0 are written while the values of TSG3nSTE and TSG3nPOT are 1, the output pattern changes. Thereafter, when a pattern switch trigger is generated on rising of the TSG3nOPC10 and TSG3nOPC11 signals, the output is switched according to the order of pattern switching. When TSG3nS120DCO = 0, the output is switched immediately. When TSG3nS120DCO = 1, the output is switched when the main counter (TSG3nCNTE) matches with TSG3nCMP0E (from the next timer cycle). TSG3nSPC2 to TSG3nSPC0 remain unchanged even if the output pattern is switched.

NOTES

- PAT0T to PAT5T: PWM output set by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, TSG3nCMP10E
- PAT0B to PAT5B: PWM output set by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, TSG3nCMP12E

Table 25.86 List of Output Patterns in 120-DC Mode (3/4)

Electric current direction: normal (TSG3nIDC = 0)
 Pattern order direction: reverse (TSG3nTSF = 1 or TSG3nPSC = 1)

| Output pins | TSG3nOPT1.TSG3nSPC2-TSG3nSPC0 ^{*1} /TSG3nSTR1.TSG3nOPF2-TSG3nOPF0 | | | | | | | |
|-------------|--|-------|-------|-------|-------|-------|-----|-----|
| | 101 | 100 | 110 | 010 | 011 | 001 | 000 | 111 |
| TSG3nO1 | PAT1T | PAT0T | PAT5T | PAT4T | PAT3T | PAT2T | Low | Low |
| TSG3nO2 | PAT1B | PAT0B | PAT5B | PAT4B | PAT3B | PAT2B | Low | Low |
| TSG3nO3 | PAT3T | PAT2T | PAT1T | PAT0T | PAT5T | PAT4T | Low | Low |
| TSG3nO4 | PAT3B | PAT2B | PAT1B | PAT0B | PAT5B | PAT4B | Low | Low |
| TSG3nO5 | PAT5T | PAT4T | PAT3T | PAT2T | PAT1T | PAT0T | Low | Low |
| TSG3nO6 | PAT5B | PAT4B | PAT3B | PAT2B | PAT1B | PAT0B | Low | Low |

Table 25.86 List of Output Patterns in 120-DC Mode (4/4)

Electric current direction: reverse (TSG3nIDC = 1)
 Pattern order direction: reverse (TSG3nTSF = 1 or TSG3nPSC = 1)

| Output pins | TSG3nOPT1.TSG3nSPC2-TSG3nSPC0 ^{*1} /TSG3nSTR1.TSG3nOPF2-TSG3nOPF0 | | | | | | | |
|-------------|--|-------|-------|-------|-------|-------|-----|-----|
| | 101 | 100 | 110 | 010 | 011 | 001 | 000 | 111 |
| TSG3nO1 | PAT4T | PAT3T | PAT2T | PAT1T | PAT0T | PAT5T | Low | Low |
| TSG3nO2 | PAT4B | PAT3B | PAT2B | PAT1B | PAT0B | PAT5B | Low | Low |
| TSG3nO3 | PAT0T | PAT5T | PAT4T | PAT3T | PAT2T | PAT1T | Low | Low |
| TSG3nO4 | PAT0B | PAT5B | PAT4B | PAT3B | PAT2B | PAT1B | Low | Low |
| TSG3nO5 | PAT2T | PAT1T | PAT0T | PAT5T | PAT4T | PAT3T | Low | Low |
| TSG3nO6 | PAT2B | PAT1B | PAT0B | PAT5B | PAT4B | PAT3B | Low | Low |

Note 1. When TSG3nSPC2 to TSG3nSPC0 are written while the values of TSG3nSTE and TSG3nPOT are 1, the output pattern changes. Thereafter, when a pattern switch trigger is generated on rising of the TSG3nOPC10 and TSG3nOPC11 signals, the output is switched according to the order of pattern switching. When TSG3nS120DCO = 0, the output is switched immediately.
 When TSG3nS120DCO = 1, the output is switched when the main counter (TSG3nCNTE) matches with TSG3nCMP0E (from the next timer cycle). TSG3nSPC2 to TSG3nSPC0 remain unchanged even if the output pattern is switched.

NOTES

- PAT0T to PAT5T: PWM output set by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, TSG3nCMP10E
- PAT0B to PAT5B: PWM output set by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, TSG3nCMP12E

(7) Operation Start Timing in 120-DC Mode

In 120-DC mode, when trigger switch control (TSG3nOPT0.TSG3nSTE=1, TSG3nOPT0.TSG3nPOT = 1) is used, pattern set with TSG3nOPT1.TSG3nSPC2 to TSG3nOPT1.TSG3nSPC0, TSG3nOPT0.TSG3nPSC, and TSG3nOPT0.TSG3nIDC can be output. However, when pattern switch control (TSG3nOPT0.TSG3nSTE=1, TSG3nOPT0.TSG3nPOT = 0) is used, the pattern of the TSG3nPTSI2 to TSG3nPTSI0 pins can be detected but the pattern order direction (TSG3nSTR1.TSG3nTSF) cannot be determined. Therefore, set the pattern order direction in TSG3nPSC when TSG3nTE is 0. The TSG3nPSC set value is loaded to TSG3nTSF, and the value can be used for the initial pattern setting.

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 0, TSG3nPOT = 0, TSG3nIDC = 0, TSG3nSTE = 1

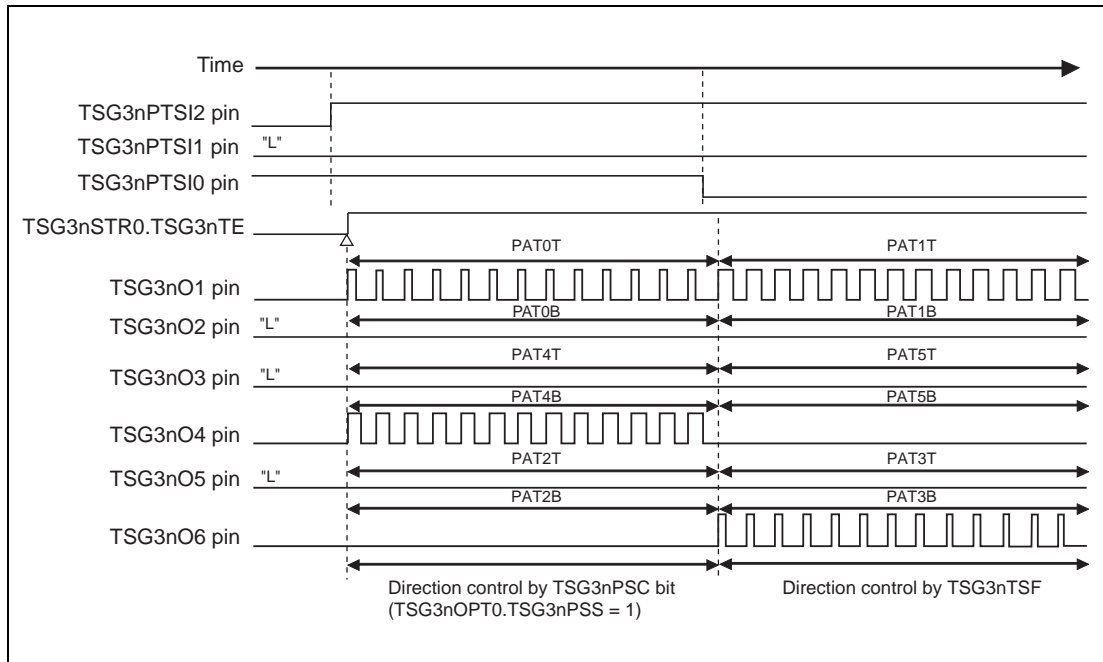


Figure 25.84 Control when Timer Output Starts in Normal Rotation (when Normal Pattern is Input)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 1, TSG3nPOT = 0, TSG3nIDC = 1, TSG3nSTE = 1

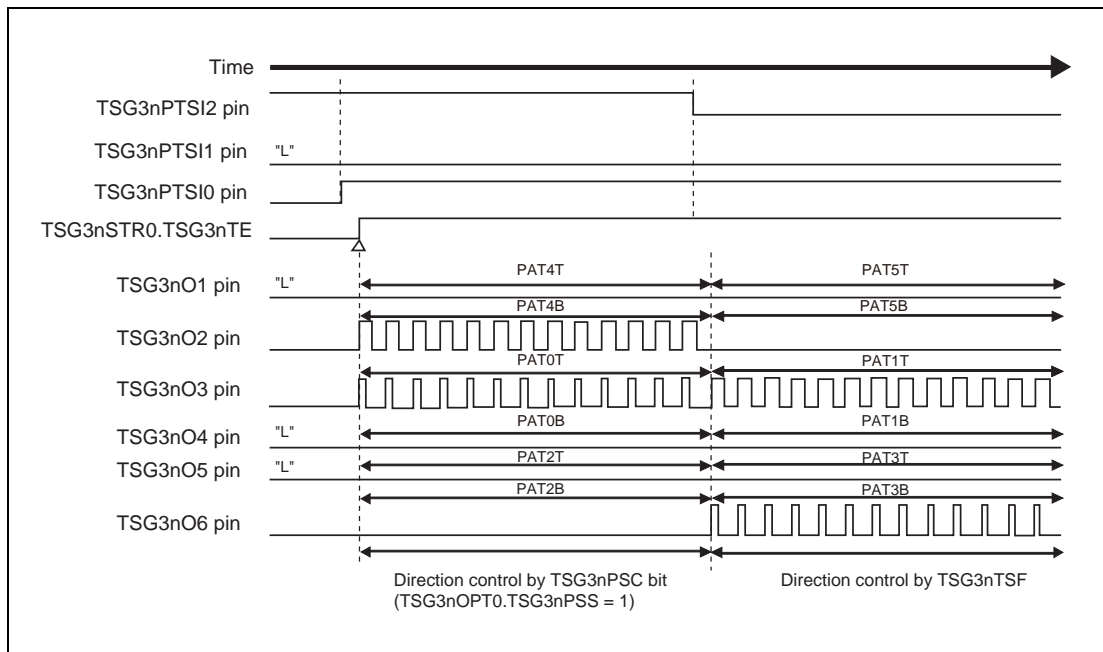


Figure 25.85 Control when Timer Output Starts in Reverse Rotation (when Normal Pattern is Input)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 0, TSG3nPOT = 0, TSG3nIDC = 0, TSG3nSTE = 1

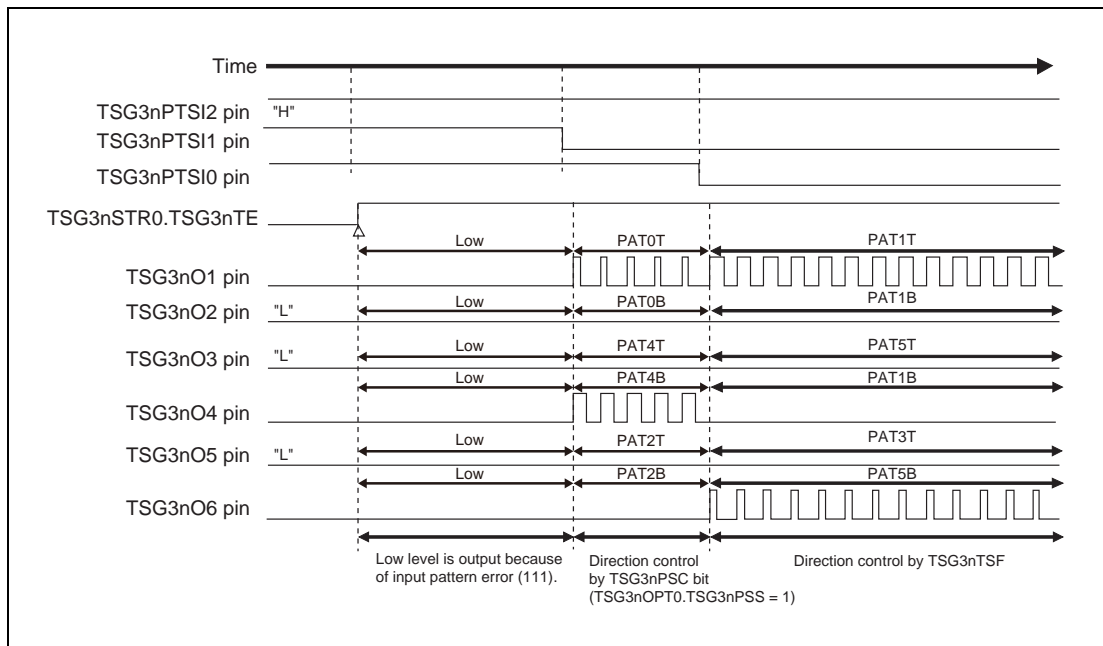


Figure 25.86 Control when Timer Output Starts in Normal Rotation (when Error Pattern is Input)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 1, TSG3nPOT = 0, TSG3nIDC = 1, TSG3nSTE = 1

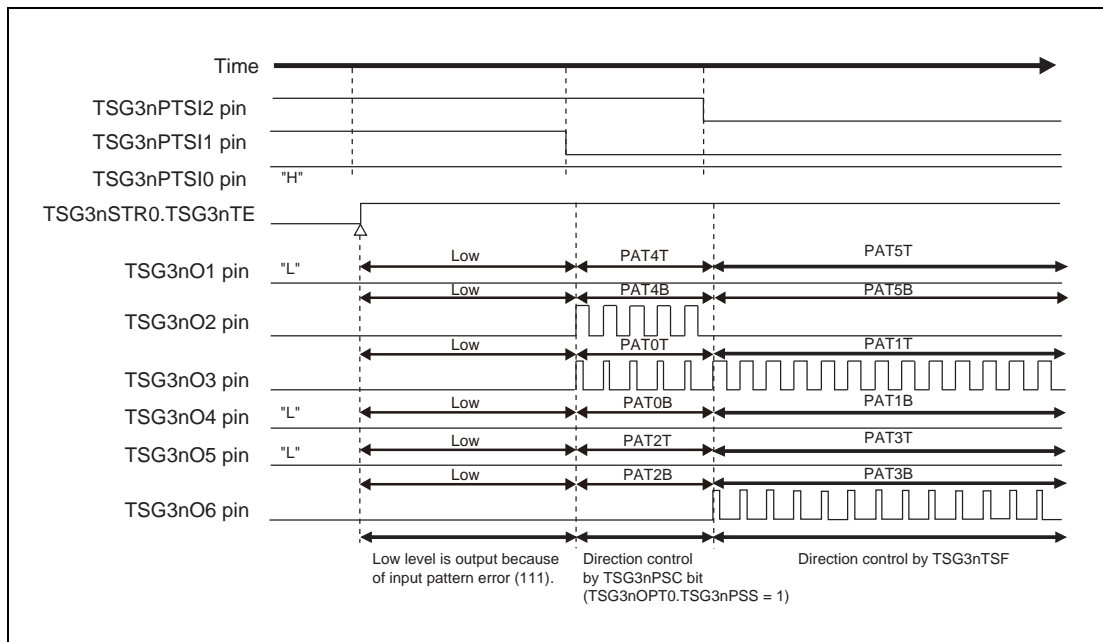


Figure 25.87 Control when Timer Output Starts in Reverse Rotation (when Error Pattern is Input)

(8) Output Switch Timing in 120-DC Mode (TSG3nS120DCO = 0)

As shown in **Figure 25.88** to **Figure 25.91**, in 120-DC mode, the external switch timing for output pattern (TSG3nOPCI0 and TSG3nOPCI1 signals, and TSG3nPTSI2 to TSG3nPTSI0 pins) is input irrespective of the 18-bit counter operation. When TSG3nS120DCO is 0, the output is switched to the new pattern by clearing the 18-bit counter using the pattern switch timing signal that is externally input.

In the pattern switch method, if a change in TSG3nPTSI2 to TSG3nPTSI0 pins occurs several times within one period, the output pattern is switched by clearing the 18-bit counter at each change. In the trigger switch method, if TSG3nOPCI0 and TSG3nOPCI1 signal trigger is input for several times within one period, the output pattern is switched by clearing the 18-bit counter each time the trigger is accepted.

If TSG3nSPC2 to TSG3nSPC0 are rewritten several times within one period, the output pattern is switched by clearing the 18-bit counter each time TSG3nSPC2 to TSG3nSPC0 are rewritten.

In case of a conflict between a rewrite to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 and TSG3nOPCI0 and TSG3nOPCI1 trigger, rewriting of TSG3nSPC2 to TSG3nSPC0 takes precedence.

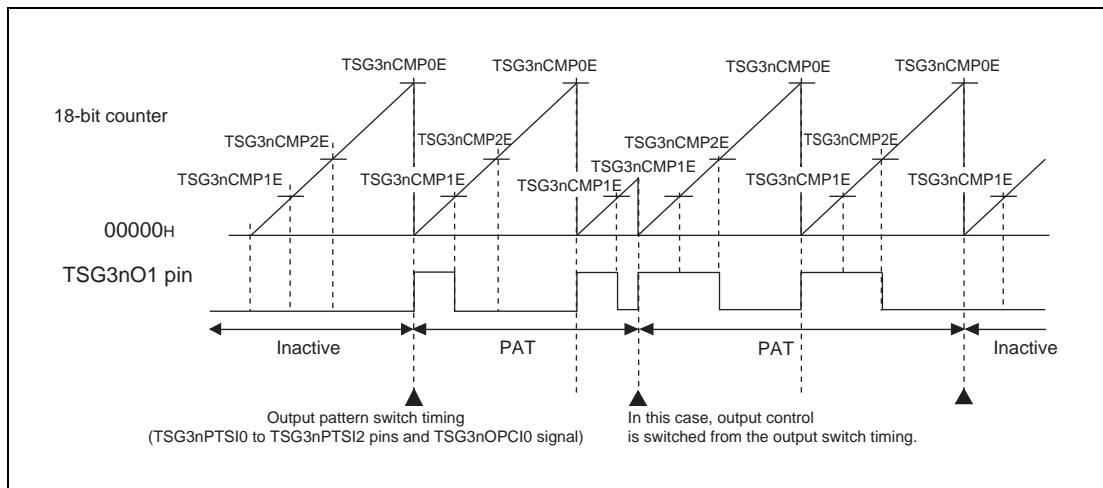


Figure 25.88 Output Switch Example (TSG3nPTSI2 to TSG3nPTSI0 Pins and TSG3nOPCI0 and TSG3nOPCI1 Signal Trigger Input)

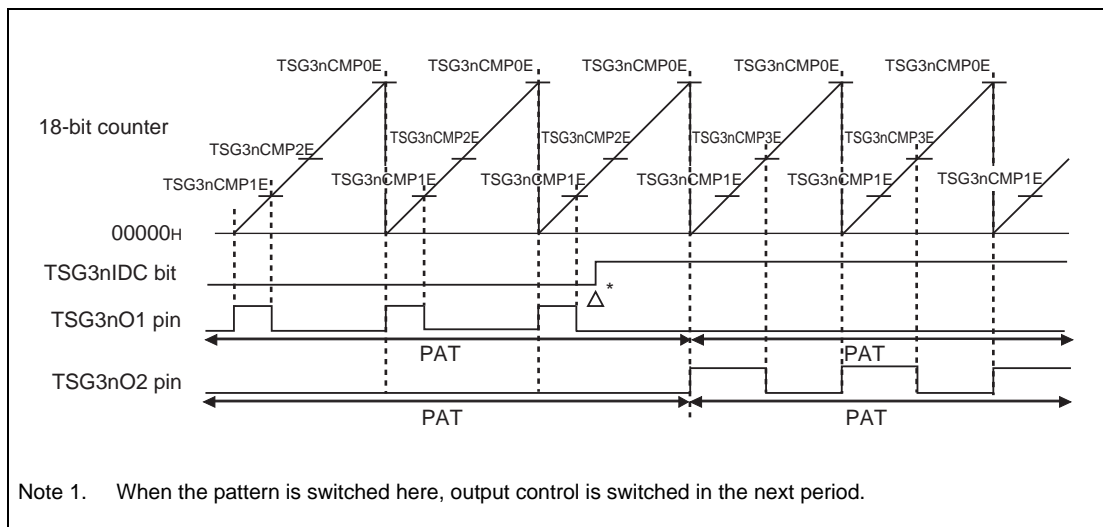


Figure 25.89 Output Switch Example (Switched by TSG3nOPT0.TSG3nIDC)

NOTE

If a change in the TSG3nPTSI2-0 pins occurs by the time next period when output control is switched by the TSG3nIDC bit, the 18-bit counter is cleared and output control is switched.

- TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1

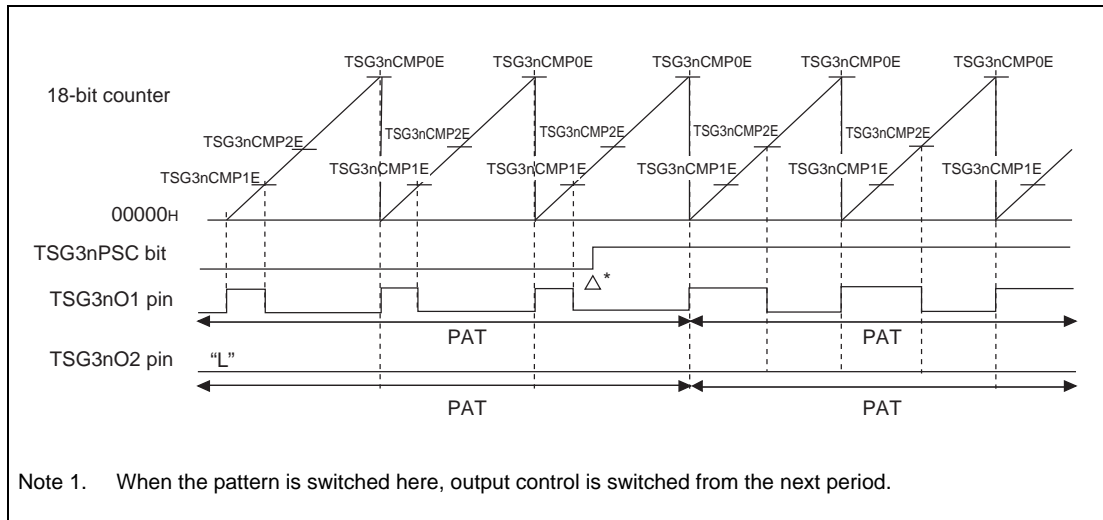


Figure 25.90 Output Switch Example (Switched by TSG3nOPT0.TSG3nPSC)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1

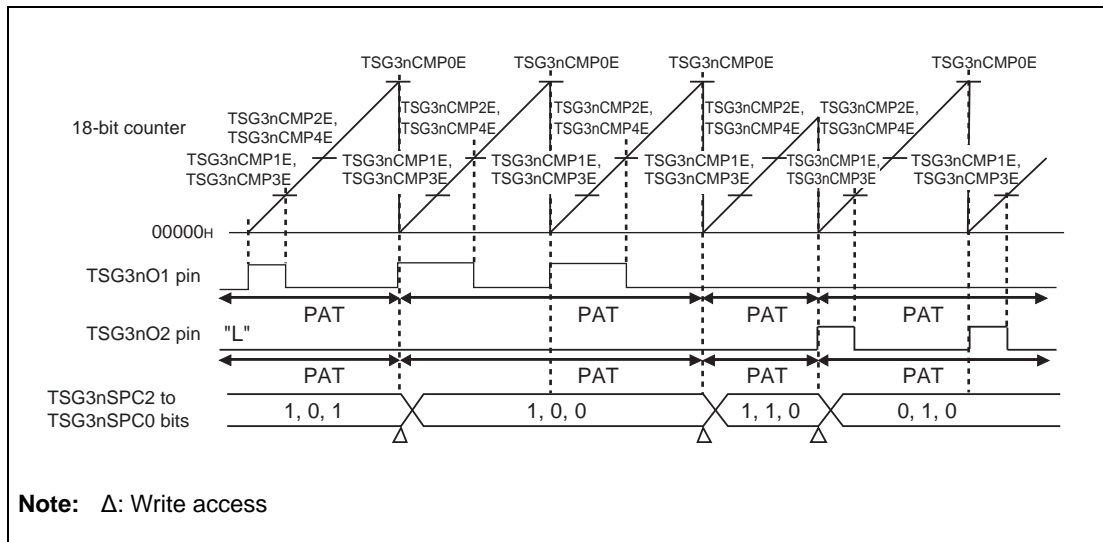


Figure 25.91 Output Switch Example (Switched by TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0)

(9) Compare Register Rewrite Timing in 120-DC Mode

Example of operation when TSG3nCMP1E is reloaded (rewritten simultaneously) is shown below.

Figure 25.92 shows an output example when TSG3nCMP1E is rewritten. After TSG3nCMP1E is changed, data is not transferred to the TSG3nCMP1E buffer register (changed data does not become valid) until the next reload timing; therefore, the specified output waveform can be obtained. However, do not write to TSG3nCMP1E again while the reload is suspended (period from when TSG3nCMP1E is changed to when simultaneous rewrite is executed). Be sure to read the reload request flag (TSG3nRSF) to confirm that the flag is 0 before writing to TSG3nCMP1E.

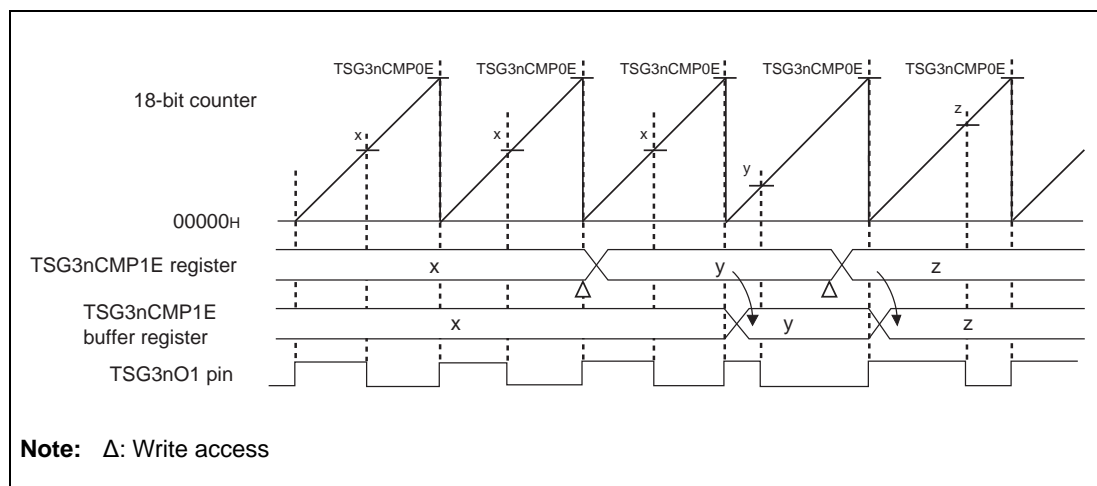


Figure 25.92 Output Example when TSG3nCMP1E is Rewritten

(10) Dead Time Control in 120-DC Mode

In 120-DC mode, the dead time is controlled on falling of each phase, and the dead time is added.

The dead time set in TSG3nDTC1W is inserted on falling of the positive phase, and the dead time set in TSG3nDTC0W is inserted on falling of the inverse phase.

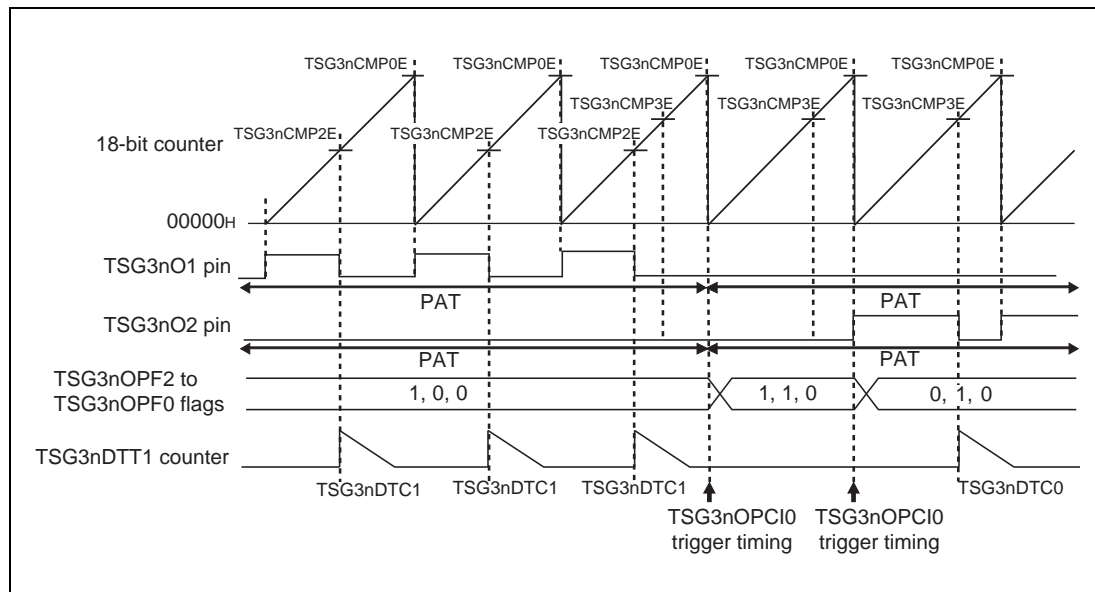


Figure 25.93 Output Switch Example

CAUTION

The dead time control method may affect the timer output. The timer output may not have the specified active level width due to the dead time control under the following conditions:

- When noise is generated on the input pattern in the pattern switch method
- When a change in the input pattern occurs earlier than the PWM period in the pattern switch method
- When TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 are changed and the output pattern is forcibly changed in the trigger switch method
- When switch method is changed
- When the current direction control bit (TSG3nOPT0.TSG3nIDC) is changed
- When the software output control function is used

(11) Output Switch in 120-DC Mode

In 120-DC mode, the output pattern can be controlled by writing values to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 when the trigger switch method (TSG3nOPT0.TSG3nSTE = 1, and TSG3nPOT = 1) is used. The dead time is secured by hardware at the switch timing.

CAUTION

When 111_B or 000_B is written to TSG3nSPC2 to TSG3nSPC0, the TSG3nO1 to TSG3nO6 pins are driven low.

(12) Operation when Noise is Generated in TSG3nPTSI2 to TSG3nPTSI0 Pins in 120-DC Mode

Input to the TSG3nPTSI2 to TSG3nPTSI0 pins is assumed to be the hall sensor signals of the brushless DC motor. Depending on the system, a noise may be generated on the TSG3nPTSI2 to TSG3nPTSI0 pins. Operation when a noise is generated is described below.

For system product design, be sure to insert a noise filter circuit between the hall sensor and the TSG3nPTSI2 to TSG3nPTSI0 pins.

Figure 25.95 shows a case when a noise is generated on the TSG3nPTSI2 to TSG3nPTSI0 pins during operation with the pattern switch method used.

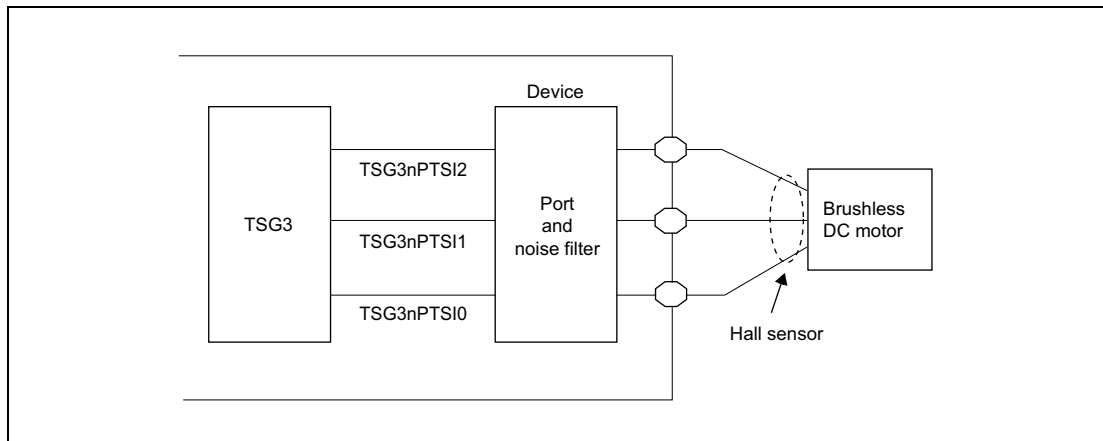


Figure 25.94 Example of Noise Filter Circuit Connection

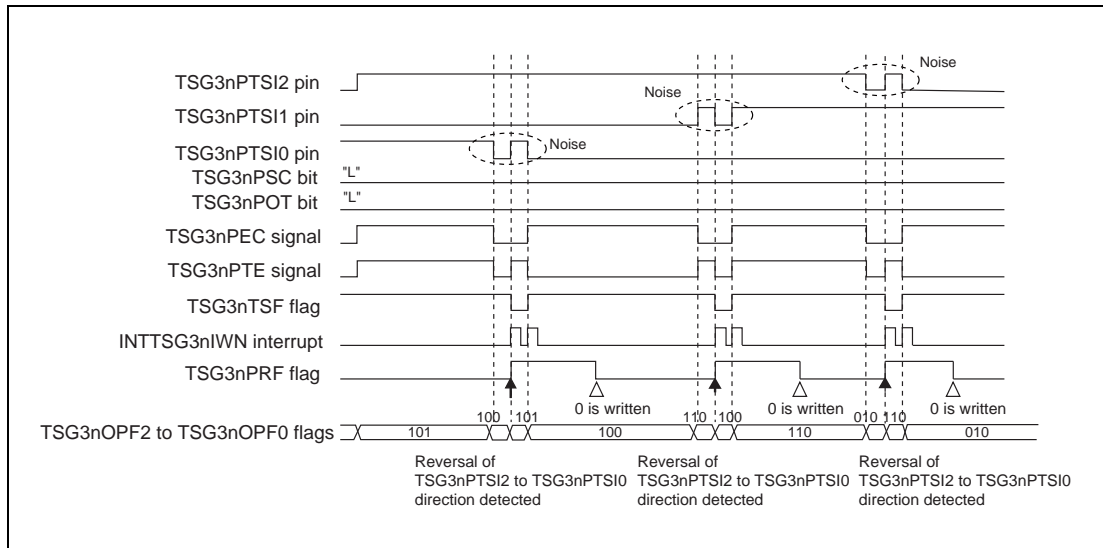


Figure 25.95 Example of Noise Generation at Level Change in TSG3nPTSI2 to TSG3nPTSI0 Pins (Pattern Switch Method)

(a) Change Timing of Input Pattern Change Detection Signal (TSG3nPTE)

The TSG3nPTE signal toggles when the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes.

CAUTION

Be sure to specify the pattern order direction with the TSG3nPSC bit in the TSG3nOPT0 register (when the TSG3nPSS bit in the TSG3nOPT0 register is 1).

When TSG3nPSC = 0:

Table 25.87 TSG3nPTE Toggle Operation when TSG3nPSC is 0

| | | TSG3nPTSI2-TSG3nPTSI0 Pins after change | | | | | | | |
|--|-----|---|-----|--------|--------|--------|--------|--------|--------|
| | | 000 | 111 | 101 | 100 | 110 | 010 | 011 | 001 |
| Current TSG3nPTSI2 to TSG3nPTSI0 pins | 000 | — | — | — | — | — | — | — | — |
| | 111 | — | — | — | — | — | — | — | — |
| | 101 | — | — | — | Toggle | — | — | — | — |
| | 100 | — | — | — | — | Toggle | — | — | — |
| | 110 | — | — | — | — | — | Toggle | — | — |
| | 010 | — | — | — | — | — | — | Toggle | — |
| | 011 | — | — | — | — | — | — | — | Toggle |
| | 001 | — | — | Toggle | — | — | — | — | — |

When TSG3nPSC = 1:

Table 25.88 TSG3nPTE Toggle Operation when TSG3nPSC is 1

| | | TSG3nPTSI2-TSG3nPTSI0 Pins after change | | | | | | | |
|--|-----|---|-----|--------|--------|--------|--------|--------|--------|
| | | 000 | 111 | 101 | 100 | 110 | 010 | 011 | 001 |
| Current TSG3nPTSI2 to TSG3nPTSI0 pins | 000 | — | — | — | — | — | — | — | — |
| | 111 | — | — | — | — | — | — | — | — |
| | 101 | — | — | — | — | — | — | — | Toggle |
| | 100 | — | — | Toggle | — | — | — | — | — |
| | 110 | — | — | — | Toggle | — | — | — | — |
| | 010 | — | — | — | — | Toggle | — | — | — |
| | 011 | — | — | — | — | — | Toggle | — | — |
| | 001 | — | — | — | — | — | — | Toggle | — |

(b) Change Timing of Three-Phase Encode Signal (TSG3nPEC)

The TSG3nPEC signal toggles when input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes

Table 25.89 TSG3nPEC Toggle Operation

| | | TSG3nPTSI2-TSG3nPTSI0 after change | | | | | | | |
|--|-----|------------------------------------|-----|--------|--------|--------|--------|--------|--------|
| | | 000 | 111 | 101 | 100 | 110 | 010 | 011 | 001 |
| Current TSG3nPTSI2 to TSG3nPTSI0 Pins | 000 | — | — | — | — | — | — | — | — |
| | 111 | — | — | — | — | — | — | — | — |
| | 101 | — | — | — | Toggle | — | — | — | Toggle |
| | 100 | — | — | Toggle | — | Toggle | — | — | — |
| | 110 | — | — | — | Toggle | — | Toggle | — | — |
| | 010 | — | — | — | — | Toggle | — | Toggle | — |
| | 011 | — | — | — | — | — | Toggle | — | Toggle |
| | 001 | — | — | Toggle | — | — | — | Toggle | — |

(c) Change Timing of TSG3nO1 to TSG3nO6 Pins

- When the pattern switch method is used, the output pattern changes when the input signal of the TSG3nPTSI2 to TSG3nPTSI0 pins*¹ changes. The output is also switched when two or more pins change simultaneously.
- When the trigger switch method is used, the output pattern changes at the rising edge of the TSG3nOPCI0 and TSG3nOPCI1 signals. The output also changes when data is written to TSG3nSPC2 to TSG3nSPC0*¹ in TSG3nOPT0.

Note 1. When the input pattern changes to 000 or 111, the TSG3nO1-TSG3nO6 pins are driven low. The output pattern of TSG3nO1-TSG3nO6 changes immediately only when TSG3nS120DCO = 0. When TSG3nS120DCO = 1, the output pattern is switched when the main counter (TSG3nCNT) matches with TSG3nCMP0E (from the next timer cycle).

(d) Change Timing of TSG3nTSF Flag

The TSG3nTSF flag toggles when the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes

Table 25.90 Setting and Clearing of TSG3nTSF

| | | TSG3nPTSI2-TSG3nPTSI0 Pins after change | | | | | | | |
|--|-----|---|-----|-----|-----|-----|-----|-----|-----|
| | | 000 | 111 | 101 | 100 | 110 | 010 | 011 | 001 |
| Current TSG3nPTSI2 to TSG3nPTSI0 Pins | 000 | — | — | — | — | — | — | — | — |
| | 111 | — | — | — | — | — | — | — | — |
| | 101 | — | — | — | 0 | — | — | — | 1 |
| | 100 | — | — | 1 | — | 0 | — | — | — |
| | 110 | — | — | — | 1 | — | 0 | — | — |
| | 010 | — | — | — | — | 1 | — | 0 | — |
| | 011 | — | — | — | — | — | 1 | — | 0 |
| | 001 | — | — | 0 | — | — | — | 1 | — |

(e) Set Timing of TSG3nNDF Flag

This flag is set when two or more pins of the TSG3nPTSI2 to TSG3nPTSI0 pins change simultaneously, and cleared when 1 is written to the TSG3nNDR bit. The TSG3nNDF flag is valid when 1 is set to the TSG3nNDC bit.

(f) Set Timing of TSG3nPRF Flag

This flag is set when the TSG3nTSF flag changes, and cleared when 1 is written to the TSG3nPRR bit. The TSG3nPRF flag is valid when 1 is set to the TSG3nPRC bit.

(g) Set Timing of TSG3nPEF Flag

This flag is set when 000 or 111 is input to the TSG3nPTSI2 to TSG3nPTSI0 pins, and cleared when 1 is written to the TSG3nPER bit. The TSG3nPEF flag is valid when the TSG3nPEC bit is set to 1.

(13) Basic Control Flow in 120-DC Mode

In 120-DC mode, there are eight control states as listed in **Table 25.91**.

When TSG3nOPT0.TSG3nSTE = 1 and TSG3nPOT = 0, the pattern switch method is used for 120-DC control. This is defined as fixed phase control. The fixed phase control should be performed considering the factors such as delay from the hall sensor and delay from sensor level detection to timer output. However, acceleration or deceleration can be performed simply by changing the PWM duty.

When TSG3nOPT0.TSG3nSTE = 1 and TSG3nPOT = 1, the trigger switch method is selected for 120-DC control. This is defined as variable phase control. With the variable phase control, the timer output pattern is set prior to the hall sensor; therefore, acceleration or deceleration control according to the phase difference can be performed. However, control is more complex than the fixed phase control because offset width and predicted value with respect to the hall sensor should be considered. For details, **Section 29.2.3.9, Three-Phase Pulse Input Control Function**.

When TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1, and TSG3nPSS = 1, the pattern order direction of the motor can be set with the TSG3nPSC bit in the TSG3nOPT0 register. Set TSG3nPSC to 0 for normal rotation and set TSG3nPSC to 1 for reverse rotation.

The TSG3nIDC bit in the TSG3nOPT0 register sets the electric current direction. If the same value as the rotation direction of the motor (TSG3nPSC set value) is set, acceleration control is set. If the different value from the rotation direction of the motor is set, deceleration control is set.

Table 25.91 Timer Control Status

| Status | TSG3nPSC in TSG3nOPT0 | TSG3nTSF in TSG3nSTR1 | TSG3nIDC in TSG3nOPT0 | TSG3nPOT in TSG3nOPT0 | Control |
|--------|-----------------------|-----------------------|-----------------------|-----------------------|--|
| A | — | 0 | 0 | 0 | Normal rotation, acceleration, and fixed phase |
| B | 0 | — | 0 | 1 | Normal rotation, acceleration, and variable phase |
| C | 0 | — | 1 | 1 | Normal rotation, deceleration, and variable phase |
| D | — | 0 | 1 | 0 | Normal rotation, deceleration, and fixed phase |
| E | — | 1 | 1 | 0 | Reverse rotation, acceleration, and fixed phase |
| F | 1 | — | 1 | 1 | Reverse rotation, acceleration, and variable phase |
| G | 1 | — | 0 | 1 | Reverse rotation, deceleration, and variable phase |
| H | — | 1 | 0 | 0 | Reverse rotation, deceleration, and fixed phase |

Generally, the state when the motor rotation stops is assumed to be the starting state and the control begins. First the fixed phase control is used to rotate the motor from the stopped state. Then, to accelerate the motor speed to the fast rotation, the variable phase control is switched on. Used in combination with encoder timer (ENCA), the variable phase control changes the timer output to a timing that is earlier than the change point (leading) of the hall sensors.

To decelerate the motor speed from fast rotation, the direction is switched to deceleration control by rewriting only TSG3nIDC in TSG3nOPT0. When the rotation count is reduced to low-speed rotation, the motor can be transitioned to the stopped state by decreasing the PWM duty.

State transition is shown in **Figure 25.96** and **Figure 25.97**.

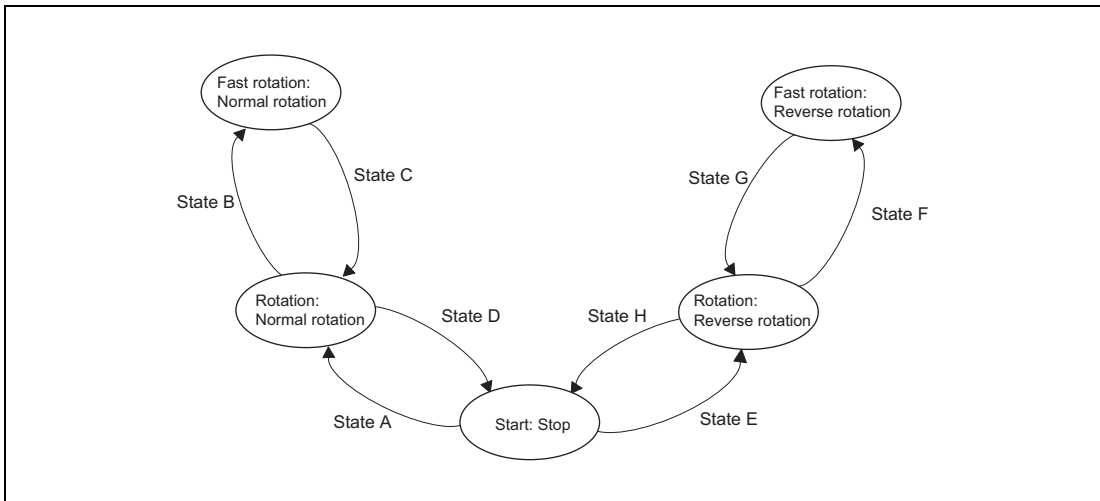


Figure 25.96 State Transition Diagram

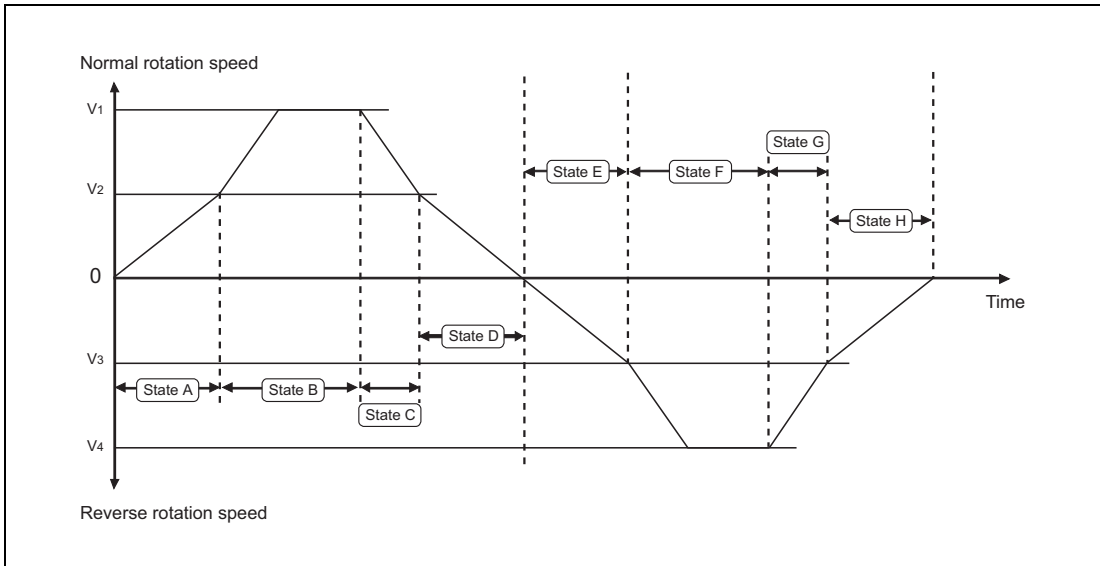


Figure 25.97 Relationship between State Transition and Rotation Speed of Motor

NOTE

V1 and V4: Fast rotation speed of normal rotation and reverse rotation
 V2 and V3: Low rotation speed of normal rotation and reverse rotation

(14) Software Output Control Function in 120-DC Mode

TSG3nOPT0.TSG3nSOC and TSG3nIDC, and TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 are used in 120-DC mode for timer output control by software.

As shown in **Figure 25.98**, the output control is switched immediately when TSG3nSOC is set to 1. If the dead time is set, the period of the dead time period is guaranteed. After that, to switch the output control from software output control to 120-DC control, set TSG3nSOC to 0. At this timing, output control is retained and at the reload timing, output control is switched to 120-DC mode.

For details on software output control function, see **Section 25.4.7.8, Software Output Control Function**.

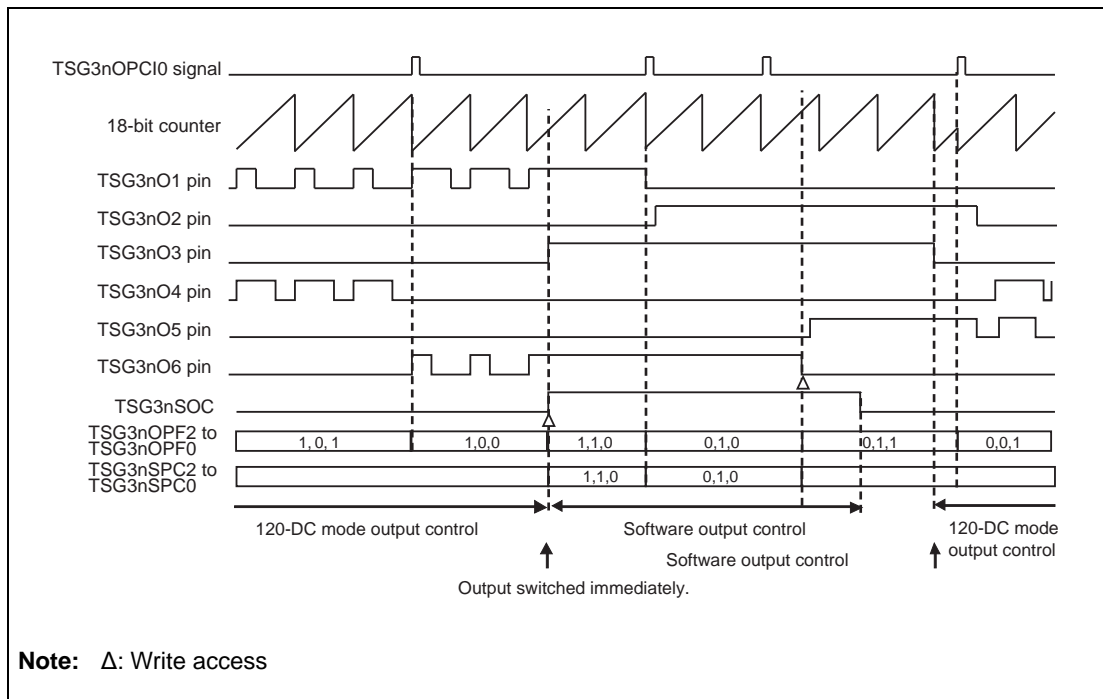


Figure 25.98 Example of Switching from 120-DC Mode to Software Output Control Function

(a) Procedure for Software Output Control

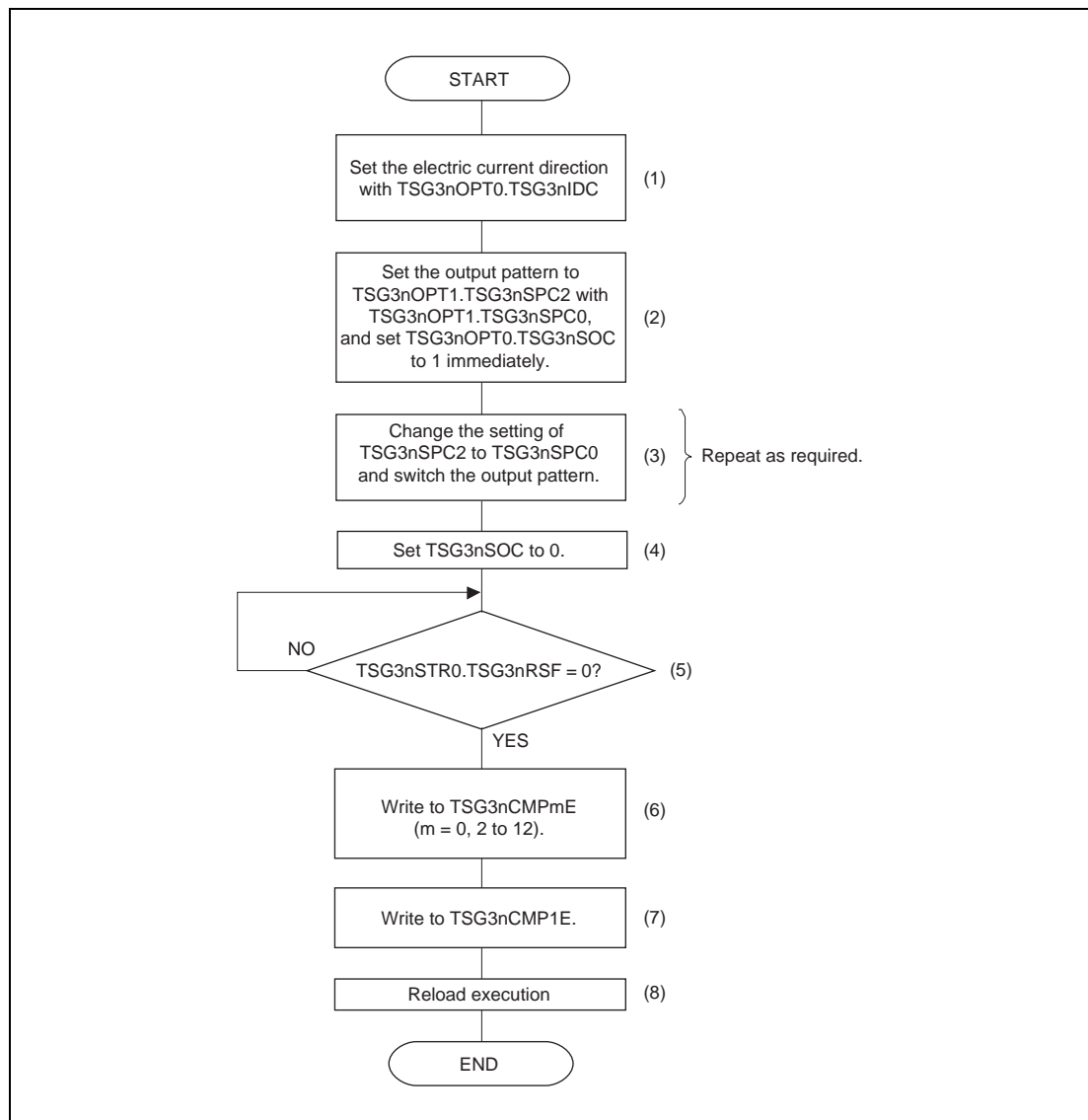


Figure 25.99 Process Flow of Software Output Control

The procedure for software output control is described below.

- (1) Set the TSG3nIDC bit. The phase of the timer output with TSG3nIDC = 0 is different by 180 degrees from that with TSG3nIDC = 1. With the software output control function, the timer output will not change only by rewriting this bit. However, if the period match occurs before step (2), the output pattern of 120-DC control changes. Therefore, specify a schedule that will prevent the period match from occurring before step (2).
- (2) Set the output pattern with TSG3nSPC2 to TSG3nSPC0. To enable software output control, set TSG3nSOC to 1 simultaneously.
- (3) Change the output pattern setting of TSG3nSPC2 to TSG3nSPC0 to change the timer output.
- (4) Confirm that the reload request flag (TSG3nRSF) = 0. If TSG3nRFS = 1, do not proceed to the following step until TSG3nRSF = 0.
- (5) By setting TSG3nSOC = 0 the software control release process starts (it is not released yet at this point).

- (6) Configure the compare register settings that are required after the software output control is released. Proceed to the following step when the register setting is not required. When setting the registers with the reload function, set them at this point.
- (7) Write to TSG3nCMP1E to start reloading.
- (8) Reload is executed and software output is released.

CAUTION

Execute reload after completing steps (4) to (7). When reload cannot be executed, the software output cannot be released.

25.4.7.5 HSP-PWM Mode (High accuracy Shifted-pulse - Pulse Width Modulation Mode)

Overview

In this mode, the 18-bit counter and the 18-bit compare register are used to generate a high accuracy sawtooth waveform PWM signal.

Prerequisites

- Set the PWM period to TSG3nCMP0E.
- Set PWM output width with the TSG3nHSPCMUE, TSG3nHSPCMVE, and TSG3nHSPCMWE registers. Set PWM shift width with the TSG3nHSPSHUE, TSG3nHSPSHVE, and TSG3nHSPSHWE registers. Set dead time with the TSG3nDTC0W and TSG3nDTC1W registers. The values set in these registers are immediately reflected to the corresponding TSG3nCMPmE (m = 1 to 12) based on the calculation described later.

Functional description

Set the PWM period, the dead time, and the PWM shift width. Then set the PWM output width. Counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter starts counting from 00000_H and is cleared by the match with TSG3nCMP0E.

During counting, a compare match interrupt (INTTSG3nI0 to INTTSG3nI12) is generated by the match of the buffer register TSG3nCMP0E to TSG3nCMP12E with the 18-bit counter.

NOTE

The HSP-PWM mode is set when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 100_B.

(1) Basic Timing Chart

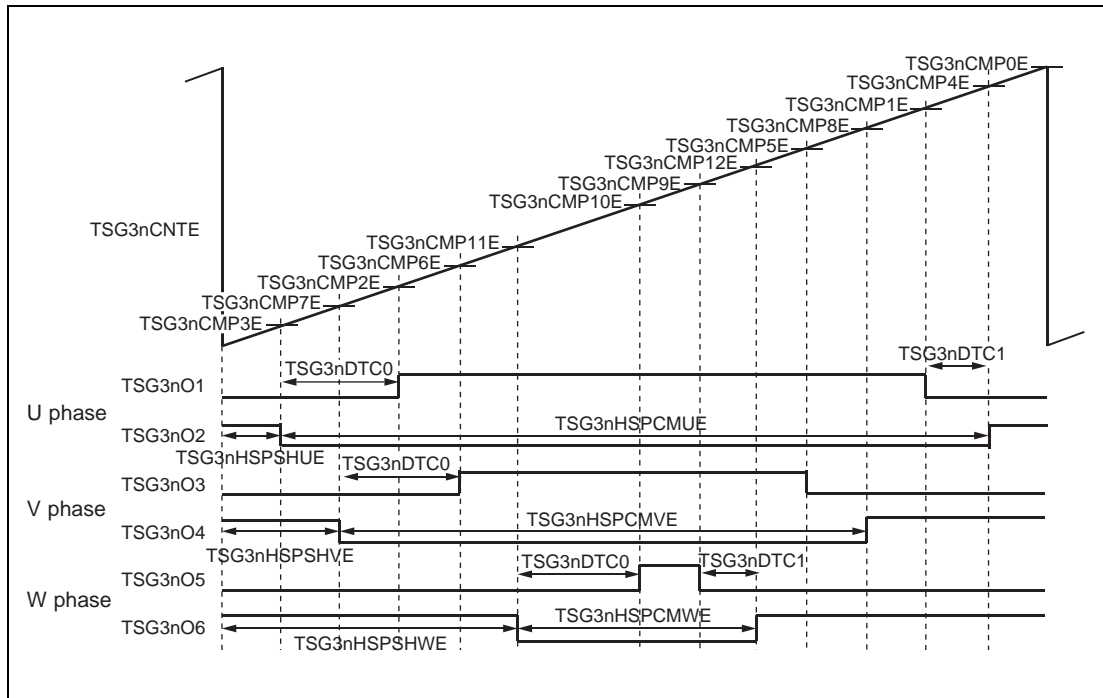


Figure 25.100 Basic Timing in HSP-PWM Mode

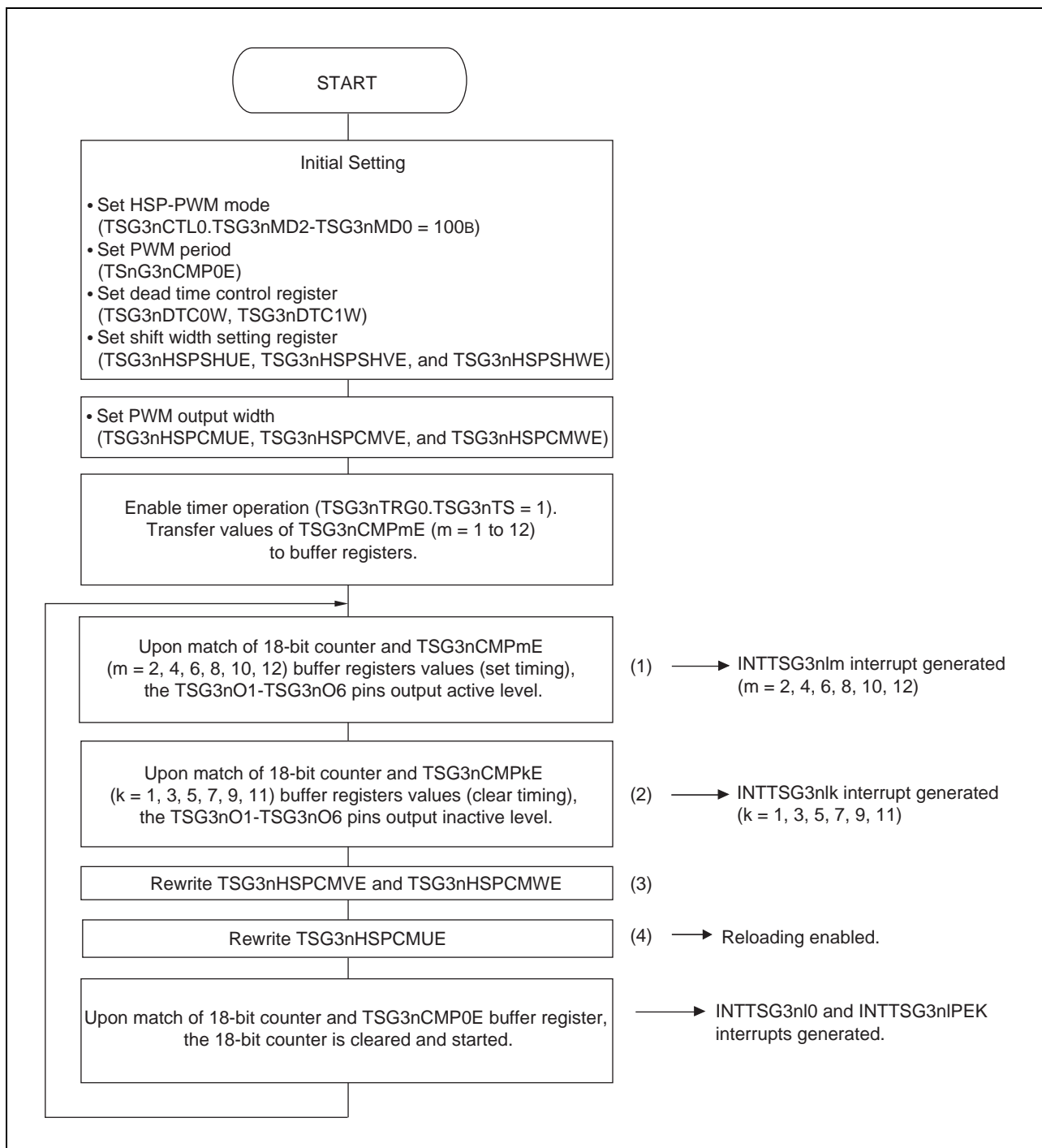


Figure 25.101 Basic Operation Flow in HSP-PWM Mode

NOTE

The timing may be different from the orders described above but be sure to execute (4) after (3).

CAUTIONS

1. When changing the settings for PWM output width in the TSG3nHSPCMUE, VE, and WE registers during operation, set the TSG3nHSPCMUE register last. When

only the settings for the V phase and W phase are changed but the PWM output width in the U phase is not, the existing value should be written back to the TSG3nHSPCMUE register.

2. After changing the PWM cycle by using the TSG3nCMP0E register, the PWM output width settings in the TSG3nHSPCMUE, VE, and WE registers must be reconfigured. Changing the settings for the PWM shift width in the TSG3nHSPSHUE, VE, and WE registers and the settings for the PWM cycle in the TSG3nCMP0E register at the same time is not allowed.
 3. Reconfigure the settings for PWM output width in the TSG3nHSPCMUE, VE, and WE registers after changing the settings for PWM shift width in the TSG3nHSPSHUE, VE, and WE registers.
-

(2) List of Operation in HSP-PWM**Table 25.92 Counter Function in HSP-PWM**

| Operation | | Setting Condition |
|----------------|-------|--|
| 18-bit counter | Start | TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger |
| | Clear | Compare match of TSG3nCMP0E buffer register and 18-bit counter |
| | Stop | TSG3nTRG1.TSG3nTT = 0 → 1 |

Table 25.93 Functions of Compare Registers, Shift Width Setting Register, and Dead Time Control Register in HSP-PWM

| Register | Rewriting Method | Rewrite during Operation | Function |
|--|------------------|--------------------------|---|
| TSG3nCMP0E | Reload | Possible | Setting period |
| TSG3nHSPCMUE TSG3nHSPSHUE | Reload | Possible | PWM control for U phase |
| TSG3nHSPCMVE TSG3nHSPSHVE | Reload | Possible | PWM control for V phase |
| TSG3nHSPCMWE TSG3nHSPSHWE | Reload | Possible | PWM control for W phase |
| TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E | Reload | Possible | Diagnostic output or A/D conversion trigger |
| TSG3nDTC0W, TSG3nDTC1W | Reload | Prohibited | Dead time |

Table 25.94 Timer Output in HSP-PWM Mode

| Pin | Function |
|---------|--|
| TSG3nO1 | PWM output by compare match of the TSG3nCMP1E buffer register (clear timing) or the TSG3nCMP2E buffer register (set timing) with the 18-bit counter. |
| TSG3nO2 | PWM output by compare match of the TSG3nCMP3E buffer register (clear timing) or the TSG3nCMP4E buffer register (set timing) with the 18-bit counter. |
| TSG3nO3 | PWM output by compare match of the TSG3nCMP5E buffer register (clear timing) or the TSG3nCMP6E buffer register (set timing) with the 18-bit counter. |
| TSG3nO4 | PWM output by compare match of the TSG3nCMP7E buffer register (clear timing) or the TSG3nCMP8E buffer register (set timing) with the 18-bit counter. |
| TSG3nO5 | PWM output by compare match of the TSG3nCMP9E buffer register (clear timing) or the TSG3nCMP10E buffer register (set timing) with the 18-bit counter. |
| TSG3nO6 | PWM output by compare match of the TSG3nCMP11E buffer register (clear timing) or the TSG3nCMP12E buffer register (set timing) with the 18-bit counter. |
| TSG3nO7 | Diagnostic signal output or pulse output by A/D conversion trigger |

Table 25.95 Interrupt Request in HSP-PWM Mode

| Interrupt | Function |
|--------------------------|---|
| INTTSG3nIm (m = 0 to 12) | Compare match of the TSG3nCMPmE buffer register with 18-bit counter |
| INTTSG3nIER | Error |
| INTTSG3nIVLY | — |
| INTTSG3nIPEK | Peak interrupt (generated at the same timing as INTTSG3nI0) |
| INTTSG3nIWN | Error |

Table 25.96 Compare Match Timing in HSP-PWM Mode

| Compare Match | Timing |
|--------------------------|---|
| TSG3nCMP0E | When 18-bit counter changes from TSG3nCMP0E to 00000 _H |
| TSG3nCMPmE (m = 1 to 12) | After match of 18-bit counter and TSG3nCMPmE |

(3) Various Settings of HSP-PWM Mode

Setting mode

HSP-PWM mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 100_B.

Setting timer output

The output pins TSG3nO1-6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.

The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

Error interrupt (INTTSG3nIER) due to the detection of the simultaneous active state of positive and inverse phases is enabled by setting TSG3nIOC1.TSG3nEOC to 1. For details, see **Section 25.4.6, Error/Warning Interrupt**.

Setting rewriting timing of register with reload function

This function is only available in reload mode. Set TSG3nCTL3.TSG3nRMC to 0.

Setting A/D conversion trigger output

The A/D conversion trigger 0 (TSG3nADTRG0 signal) is set with TSG3nCTL5.TSG3nAT09-TSG3nAT00.

TSG3nAT09-TSG3n00 is used to enable or disable the A/D conversion trigger output upon match of TSG3nDCMP2E-0E with 18-bit counter (up count).

TSG3nCTL6.TSG3nAT19-TSG3nAT10 is used to set the A/D conversion trigger 1 (TSG3nADTRG1 signal).

To set the match timing of the 18-bit counter and TSG3nDCMP2E-TSG3nDCMP0E, set the compare value to each register.

The skipping function can be used for TSG3nADTRG0, TSG3nADTRG1 signals. TSG3nACC01 and TSG3nACC00 of TSG3nCTL5, and TSG3nACC11 and TSG3nACC10 of TSG3nCTL6 can be used to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

- Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E-0E correctly when the timing pulse of A/D conversion trigger is output to TSG3nO7.
- In HSP-PWM mode, no valley interrupt (INTTSG3nIVLY) is generated. Therefore, TSG3nCTL5.TSG3nAT00 and TSG3nCTL6.TSG3nAT10 must be set to 0.
- In HSP-PWM mode, the 18-bit sub-counter does not operate. Therefore, TSG3nCTL5.TSG3nAT09 and TSG3nAT08, and TSG3nCTL6.TSG3nAT19 and TSG3nAT18 must be set to 0.
- In HSP-PWM mode, down counting by the 18-bit counter is not generated. Therefore, TSG3nCTL5.TSG3nAT07, TSG3nAT05, and TSG3nAT03, and TSG3nCTL6.TSG3nAT17, TSG3nAT15, and TSG3nAT13 must be set to 0.

Setting PWM period

Set the PWM period with TSG3nCMP0E.

The PWM period is calculated by the following expression:

$$\text{PCLK} \times (\text{TSG3nCMP0E} + 1)$$

Setting PWM output width

The PWM output width is set with TSG3nHSPCMUE, TSG3nHSPCMVE, and TSG3nHSPCMWE (TSG3nCMP1E-12E).

Satisfy the following requirements when setting the TSG3nHSPCMUE, TSG3nHSPCMVE, and TSG3nHSPCMWE registers:

$$0 \leq \text{TSG3nHSPCMUE}, \text{TSG3nHSPCMVE}, \text{TSG3nHSPCMWE} \leq \text{TSG3nCMP0E} + \text{TSG3nDTC0} + \text{TSG3nDTC1} + 1$$

Set the PWM output width after setting TSG3nCMP0E, TSG3nHSPSHUE, TSG3nHSPSHVE, TSG3nHSPSHWE, TSG3nDTC0, and TSG3nDTC1.

Setting PWM shift width

PWM shift width is set with TSG3nHSPSHUE, TSG3nHSPSHVE, and TSG3nHSPSHWE.

Satisfy the following requirements when setting TSG3nHSPSHUE, TSG3nHSPSHVE, and TSG3nHSPSHWE.

$$\text{TSG3nHSPSHUE}, \text{TSG3nHSPSHVE}, \text{TSG3nHSPSHWE} \leq \text{TSG3nCMP0E}$$

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions:

$$\text{PCLK} \times \text{TSG3nDTC0}$$

$$\text{PCLK} \times \text{TSG3nDTC1}$$

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the inactive state and a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

Satisfy the following requirements when setting TSG3nDTC0 and TSG3nDTC1.

$$(\text{TSG3nCMP0E} + \text{TSG3nDTC0} + \text{TSG3nDTC1} + 1) < 3\text{FFFF}_H$$

$$\text{TSG3nCMP0E} > 3 \times \text{TSG3nDTC0}$$

$$\text{TSG3nCMP0E} > 3 \times \text{TSG3nDTC1}$$

CAUTION

Do not modify the settings of TSG3nDTC0 and TSG3nDTC1 during timer operation in HSP-PWM (TSG3nTE = 1). Set TSG3nDTC0 and TSG3nDTC1 while TSG3nTE = 0.

Set dead time in HSP-PWM mode. Do not set 0 to TSG3nDTC0 and TSG3nDTC1.

Settings prohibited in HSP-PWM Mode

In HSP-PWM mode, use the following registers and bits with the settings shown below.

Do not modify the settings during operation (TSG3nTE = 1).

Table 25.97 Setting Prohibited in HSP-PWM Mode

| Bit Name | Setting Value | Description |
|-----------------------|----------------------|---|
| TSG3nCTL3.TSG3nRMC | 0 | Available only in reload mode. |
| TSG3nIOC3.TSG3nTOL6-1 | 000000 _B | Logical inverse of set/clear of PWM is prohibited (HSP-PWM mode limitation) |
| TSG3nOPT0.TSG3nSOC | 0 | Switching to software control function is prohibited. |
| TSG3nOPT0.TSG3nSTE | 0 | Operation setting of 120-DC mode and software control function (value after reset). |
| TSG3nOPT0.TSG3nPOT | 0 | Operation setting of 120-DC mode and software control function (value after reset). |
| TSG3nOPT0.TSG3nPSS | 0 | Operation setting of 120-DC mode and software control function (value after reset). |
| TSG3nOPT0.TSG3nIDC | 0 | Operation setting of 120-DC mode and software control function (value after reset). |
| TSG3nOPT0.TSG3nPSC | 0 | Operation setting of 120-DC mode and software control function (value after reset). |
| TSG3nOPT1.TSG3nSPC2-0 | 000 _B | Operation setting of 120-DC mode and software control function (value after reset). |
| TSG3nPAT0W | 0000000 _H | Operation setting of 120-DC mode (value after reset). |
| TSG3nPAT1W | 0000000 _H | Operation setting of 120-DC mode (value after reset). |

CAUTION

In HSP-PWM mode, no setting should be made directly to the TSG3nCMPmE register (m = 1 to 12).

The PWM output width and the PWM shift width should be set with TSG3nHSPCMUE, VE, and WE registers and TSG3nHSPSHUE, VE, and WE registers.

25.4.7.6 Compare Register Setting in HSP-PWM Mode

In HSP-PWM mode, the PWM output width is set with TSG3nHSPCMUE, VE and WE.

With a write access to TSG3nHSPCMUE, VE, and WE, TSG3 calculates and sets the values to TSG3nCMP1E-12E based on the values written to the followings:

- TSG3nCMP0E (PWM period setting)
- TSG3nDTC0 (dead time control 0)
- TSG3nDTC1 (dead time control 1)
- TSG3nHSPSHUE/VE/WE (PWM shift width setting)
- TSG3nHSPCMUE/VE/WE (PWM output width setting)

The high accuracy PWM is realized by these values.

The algorithm to set compare register values are listed in the following table.

Table 25.98 Algorithm for Compare Setting in HSP-PWM Mode

| Value Set In HSPCMUE | | | | CMP4E | CMP3E | CMP2E | CMP1E |
|-----------------------------------|---|---------|------------------------------|---|---|-----------------------|--|
| HSPCMUE = 0 | | | | if (HSPSHUE = 0) 0 else HSPCMUE - 1 + HSPSHUE | CMP0E + 1 | 0 | 0 |
| 0 | < | HSPCMUE | ≤ DTC0 + DTC1 | HSPCMUE - 1 + HSPSHUE | if (HSPSHUE = 0) CMP0E else HSPSHUE - 1 | 0 | 0 |
| DTC0 + DTC1 | < | HSPCMUE | ≤ CMP0E | HSPCMUE - 1 + HSPSHUE | if (HSPSHUE = 0) CMP0E else HSPSHUE - 1 | DTC0 - 1 + HSPSHUE | HSPCMUE - DTC1 - 1 + HSPSHUE |
| CMP0E | < | HSPCMUE | ≤ CMP0E + DTC1 + 1 | 0 | 0 | DTC0 - 1 + HSPSHUE | HSPCMUE - DTC1 - 1 + HSPSHUE |
| CMP0E + DTC1 + 1 | < | HSPCMUE | < CMP0E + DTC0 + DTC1 + 1 | 0 | 0 | DTC0 - 1 + HSPSHUE | HSPCMUE - CMP0E - DTC1 - 2 + HSPSHUE |
| HSPCMUE = CMP0E + DTC0 + DTC1 + 1 | | | | 0 | 0 | DTC0 - 1 + HSPSHUE | CMP0E + 1 |

: For the colored cells, subtract CMP0E + 1 when the calculated result is greater than CMP0E.

NOTE

“TSG3n” is omitted from the register names used in the calculation.

25.4.7.7 Timer Output Operation in HSP-PWM Mode

TSG3nO1-6 output is set by compare match of TSG3nCNTE with TSG3nCMP2E, 4E, 6E, 8E, 10E, and 12E, respectively, and cleared by compare match of TSG3nCNTE with TSG3nCMP1E, 3E, 5E, 7E, 9E, and 11E, respectively.

When PWM output width is set in TSG3nHSPCMUE, VE, and WE, the value is set in the TSG3nCMP1E-12E registers based on the calculation described in **Section 25.4.7.6, Compare Register Setting in HSP-PWM Mode** that enables a high accuracy PWM output from 0% to 100%.

PWM output timing can be shifted by setting desired width to TSG3nHSPSHUE, VE, WE.

(1) When TSG3nHSPCMUE, VE, and WE (PWM output width setting) is set to 0

When 0 is set to TSG3nHSPCMUE (U phase PWM output width setting), 0 is set to TSG3nCMP1E, 2E, and 4E, and “TSG3nCMP0E+1” is set to TSG3nCMP3E.

Setting of TSG3nO1 by the match of TSG3nCNTE with TSG3nCMP2E and clearing by the match with TSG3nCMP1E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO1 output is fixed to inactive.

Setting of TSG3nO2 by the match of TSG3nCNTE with TSG3nCMP4E occurs when TSG3nCNTE = 0. Since TSG3nCMP0E + 1 is set to TSG3nCMP3E, TSG3nCNTE does not match TSG3nCMP3E. Therefore, TSG3nO2 output is fixed to active.

The same operation applies to TSG3nO3-6, which are the output pins for V phase and W phase when 0 is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

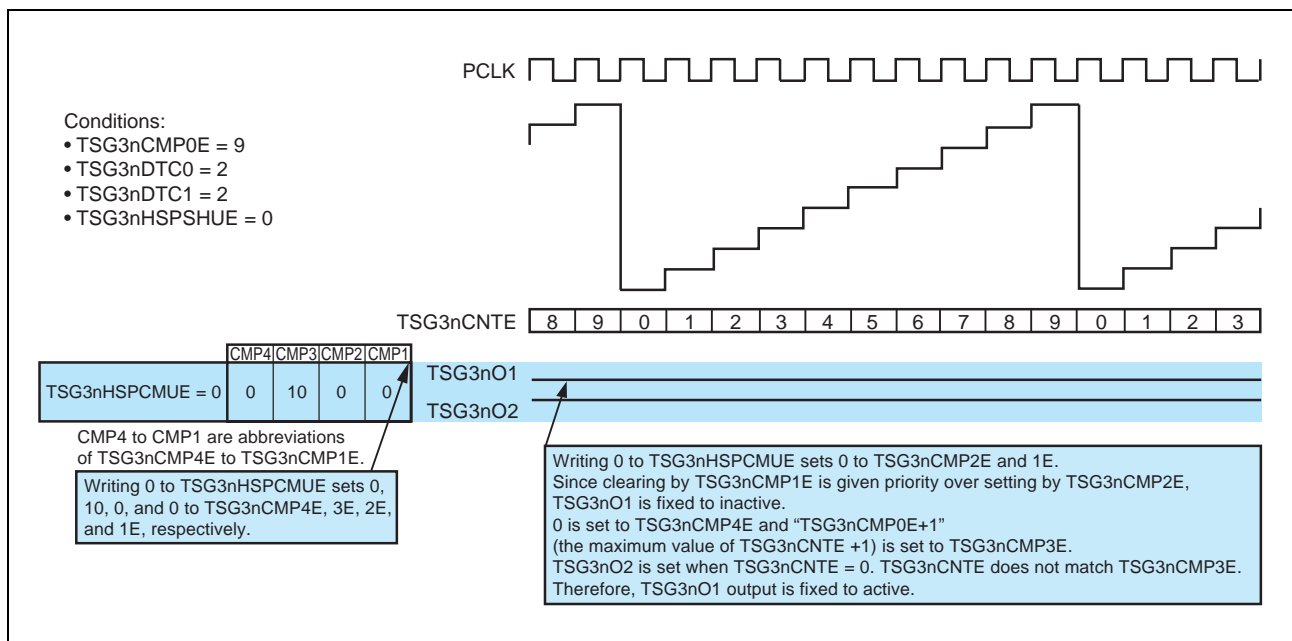


Figure 25.102 Waveform in HSP-PWM Mode (0 is set to TSG3nHSPCMUE)

(2) When TSG3nHSPCMUE/VE/WE (PWM output width setting) is set to the range $0 < \text{TSG3nHSPCMUE} \leq \text{TSG3nDTC0} + \text{TSG3nDTC1}$

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), 0 is set to TSG3nCMP1E and 2E, “TSG3nHSPCMUE set value - 1” is set to TSG3nCMP4E, and the value of the TSG3nCMP0E is set to TSG3nCMP3E.

Setting of TSG3nO1 by the match of TSG3nCNTE with TSG3nCMP2E and clearing by the match with TSG3nCMP1E occurs simultaneously, but clearing takes precedence. Therefore, TSG3nO1 output is fixed to inactive.

TSG3nO2 is cleared by the match of TSG3nCNTE with TSG3nCMP3E and set by the match with TSG3nCMP4E. Here, the output is shifted according to the set value in TSG3nHSPCMUE, that is, TSG3nO2 is inactive for one cycle during PWM period when 1 is set, two cycles when 2 is set, and three cycles when 3 is set, and so on.

When the value other than 0 is set to TSG3nHSPSHUE (PWM shift width), set/clear timing of TSG3nO2 is shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “ $0 < \text{TSG3nHSPCMUE} \leq \text{TSG3nDTC0} + \text{TSG3nDTC1}$ ” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

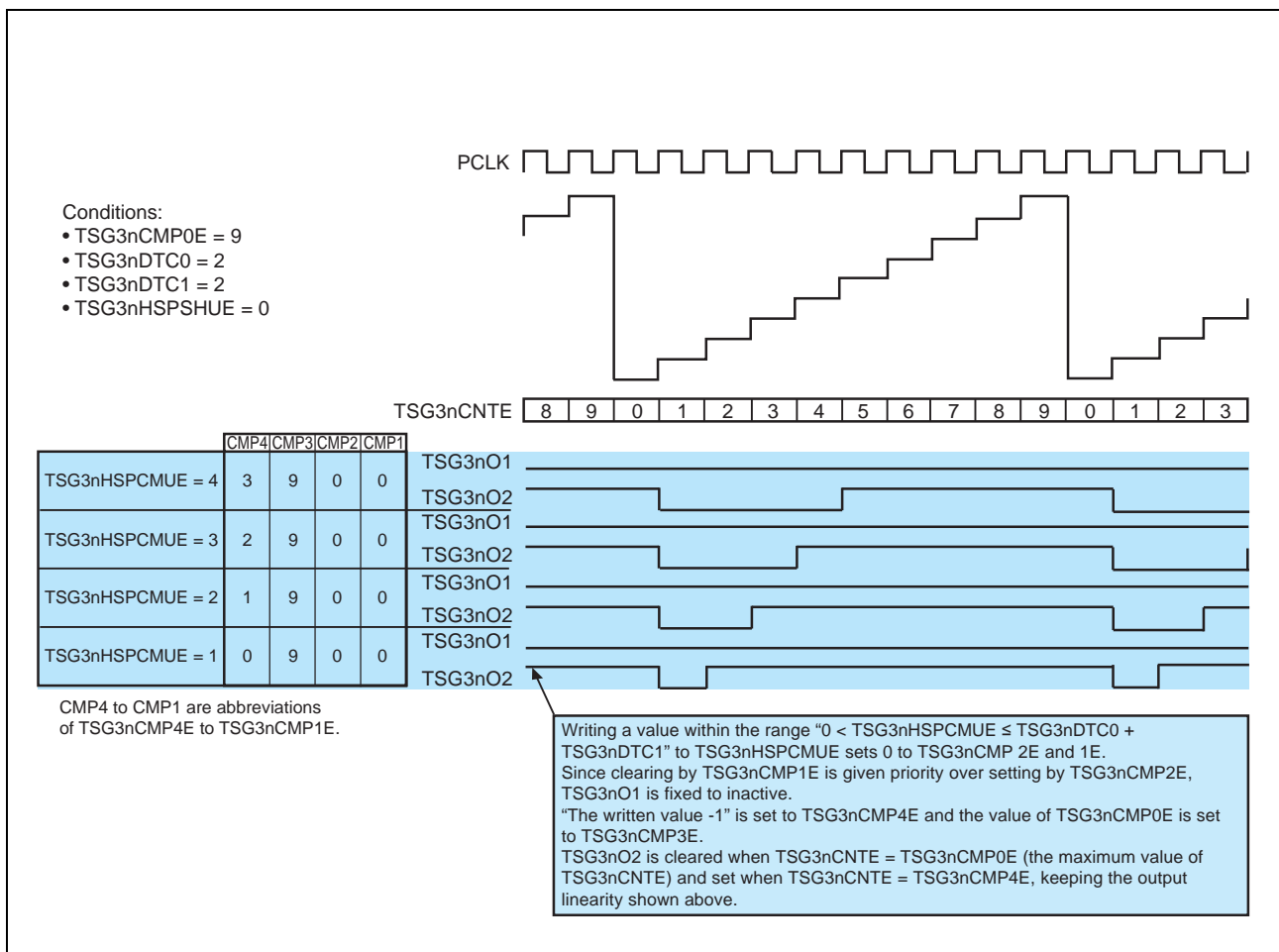


Figure 25.103 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to $0 < \text{TSG3nHSPCMUE} \leq \text{TSG3nDTC0} + \text{TSG3nDTC1}$)

(3) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to the range $TSG3nDTC0 + TSG3nDTC1 < TSG3nHSPCMUE/VE/WE \leq TSG3nCMP0E$

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), “TSG3nDTC0 - 1” is set to TSG3nCMP2E, “TSG3nHSPCMUE - TSG3nDTC1 - 1” is set to TSG3nCMP1E, “TSG3nHSPCMUE - 1” is set to TSG3nCMP4E, and the value of TSG3nCMP0E is set to TSG3nCMP3E.

TSG3nO1 is set by the match of TSG3nCNTE with TSG3nCMP2E and cleared by the match with TSG3nCMP1E while TSG3nO2 is set by the match of TSG3nCNTE with TSG3nCMP4E and cleared by the match with TSG3nCMP3E

Here, the outputs are shifted according to the set value in TSG3nHSPCMUE, that is, when “TSG3nDTC0 + TSG3nDTC1 + 1” is set, TSG3nO1 is active for one cycle during PWM period and TSG3nO2 is inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1 + 1” during PWM period. When “TSG3nDTC0 + TSG3nDTC1 + 2” is set, TSG3nO1 is active for two cycles during PWM period and TSG3nO2 is inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1 + 2” during PWM period, and so on.

When the value other than 0 is set to TSG3nHSPSHUE (PWM shift width), set/clear timing of TSG3nO1 and TSG3nO2 are shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “TSG3nDTC0 + TSG3nDTC1 < TSG3nHSPCMUE/VE/WE ≤ TSG3nCMP0E” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

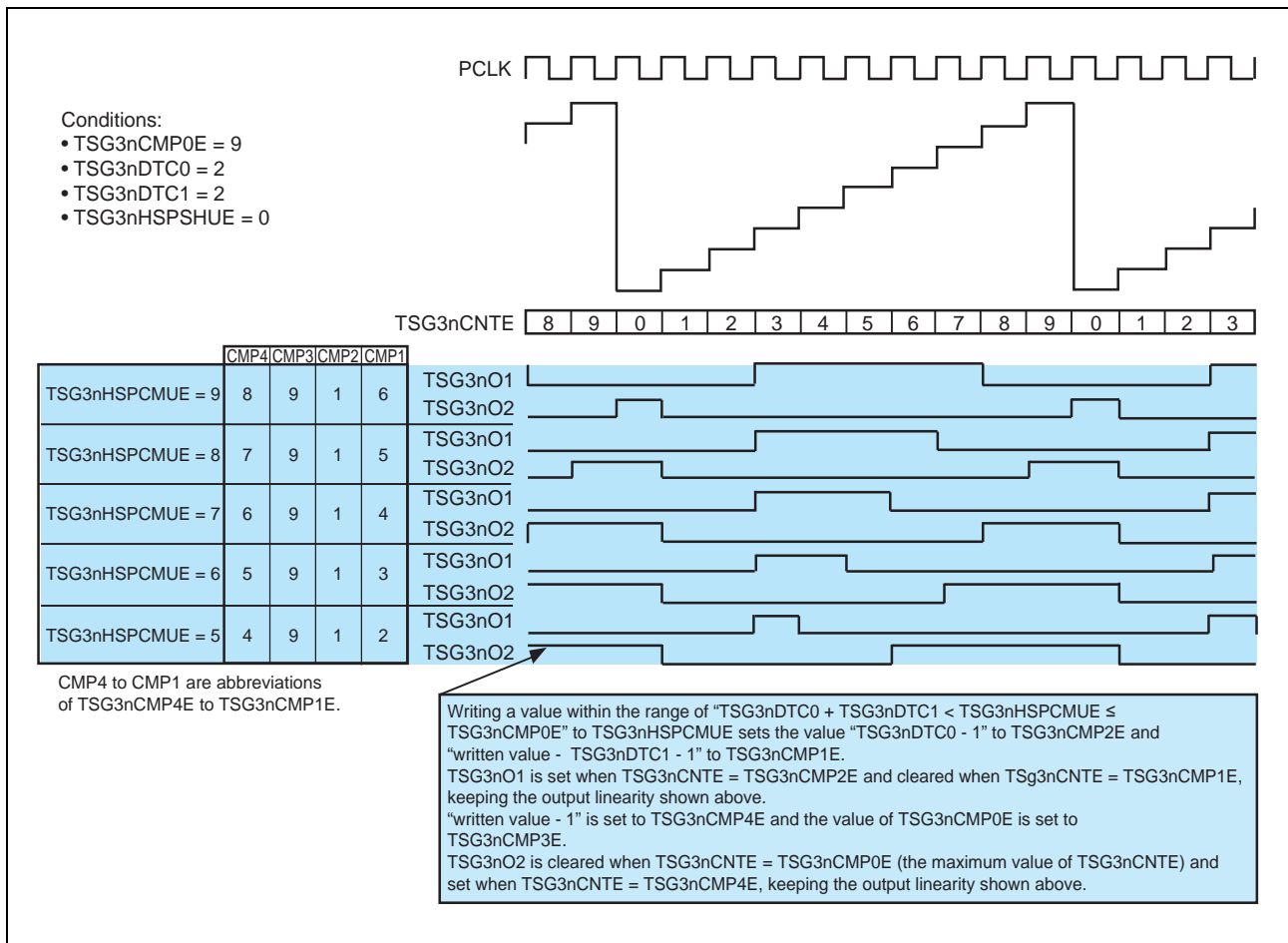


Figure 25.104 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to $TSG3nDTC0 + TSG3nDTC1 < TSG3nHSPCMUE \leq TSG3nCMP0E$)

(4) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to the range TSG3nCMP0E < TSG3nHSPCMUE/VE/ WE < TSG3nCMP0E + TSG3nDTC1 + 1

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), “TSG3nDTC0 - 1” is set to TSG3nCMP2E, “TSG3nHSPCMUE - TSG3nDTC1 - 1” is set to TSG3nCMP1E, and 0 is set to TSG3nCMP3E and 4E.

TSG3nO1 is cleared by the match of TSG3nCnTE with TSG3nCMP1E and set by the match with TSG3nCMP2E.

Here, the output is shifted according to the set value in TSG3nHSPCMUE, that is, TSG3nO1 is inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1” during PWM period when “TSG3nCMP0E + 1” is set, inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1 - 1” during PWM period when “TSG3nCMP0E + 2” is set, and so on.

Setting of TSG3nO2 by the match of TSG3nCnTE with TSG3nCMP4E and clearing by the match with TSG3nCMP3E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO2 output is fixed to inactive.

When the value other than 0 is set to TSG3nHSPSHUE (U phase PWM shift width), set/clear timing of TSG3nO1 is shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “TSG3nCMP0E < TSG3nHSPCMUE/VE/ WE < TSG3nCMP0E + TSG3nDTC1 + 1” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

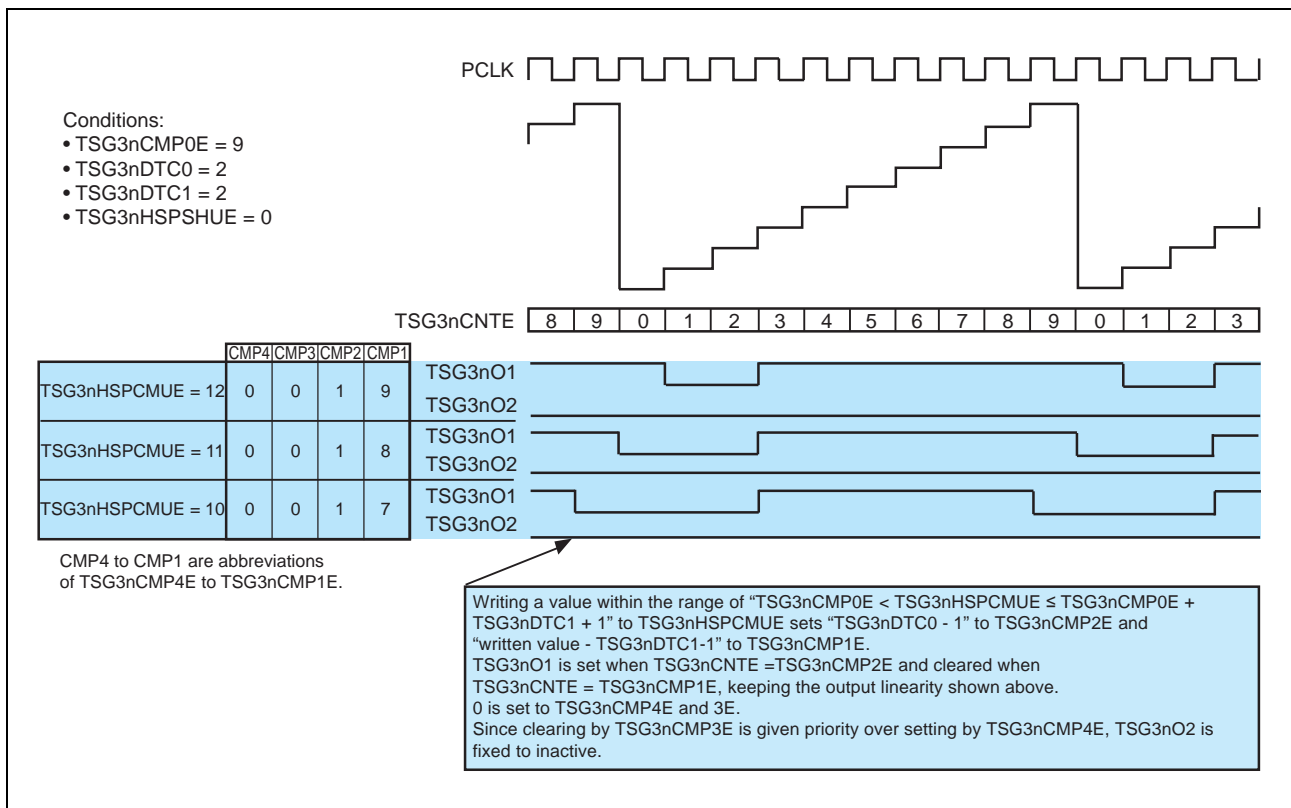


Figure 25.105 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to TSG3nCMP0E < TSG3nHSPCMUE < TSG3nCMP0E + TSG3nDTC1 + 1)

(5) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to the range $TSG3nCMP0E + TSG3nDTC1 + 1 < TSG3nHSPCMUE/VE/WE < TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1$

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), “TSG3nDTC0 - 1” is set to TSG3nCMP2E, “TSG3nHSPCMUE - TSG3nCMP0E - TSG3nDTC1 - 2” is set to TSG3nCMP1E, and 0 is set to TSG3nCMP3E and 4E.

TSG3nO1 is cleared by the match of TSG3nCnTE with TSG3nCMP1E and set by the match with TSG3nCMP2E. Here, the output is shifted according to the set value in TSG3nHSPCMUE, that is, TSG3nO1 is inactive for one cycle during PWM period when “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1” (the maximum value of PWM output width - 1) is set, inactive for two cycles during PWM period when “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 - 1” (the maximum value of PWM output width - 2) is set, and so on.

Setting of TSG3nO2 by the match of TSG3nCnTE with TSG3nCMP4E and clearing by the match with TSG3nCMP3E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO2 output is fixed to inactive.

When the value other than 0 is set to TSG3nHSPSHUE (U Phase PWM shift width), set/clear timing of TSG3nO1 is shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “TSG3nCMP0E + TSG3nDTC1 + 1 < TSG3nHSPCMUE/VE/WE < TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

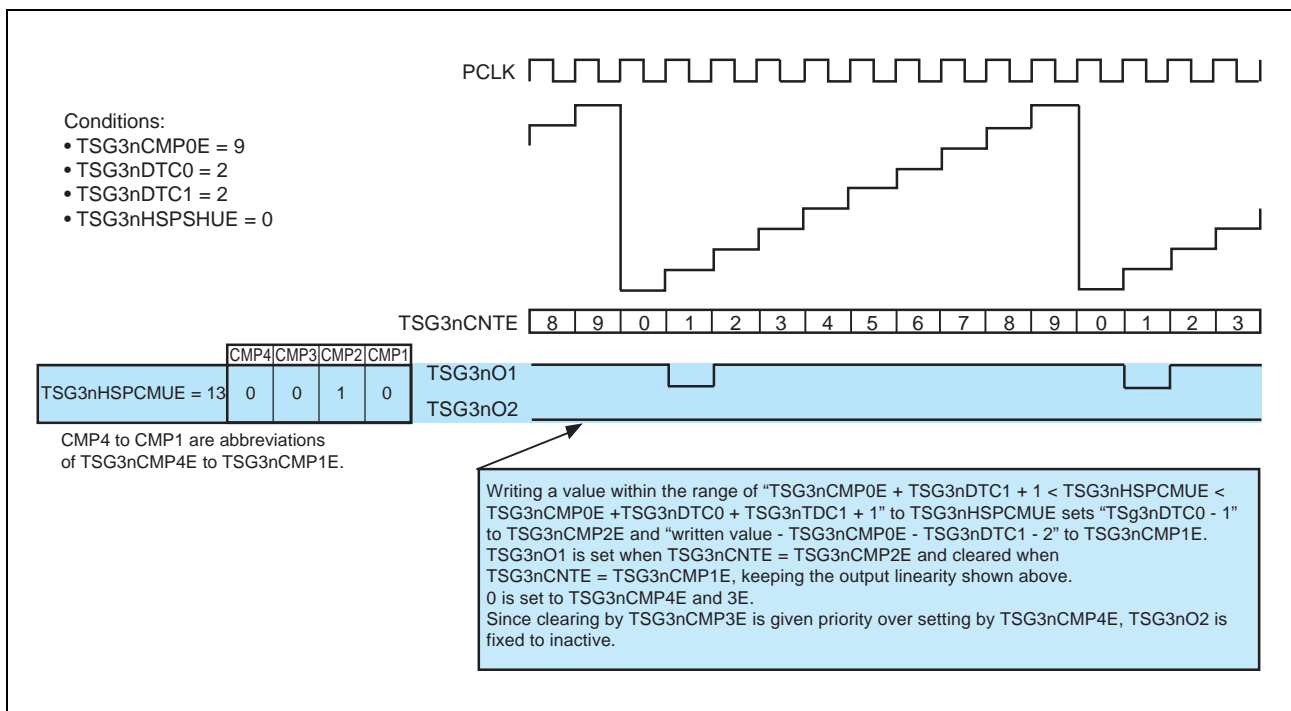


Figure 25.106 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to $TSG3nCMP0E + TSG3nDTC1 + 1 < TSG3nHSPCMUE < TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1$)

(6) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1

When “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1” (the maximum value of PWM output width) is set to TSG3nHSPCMUE, “TSG3nDTC0 - 1” is set to TSG3nCMP2E, “TSG3nCMP0E + 1” is set to TSG3nCMP1E, and 0 is set to TSG3nCMP3E and 4E.

Setting of TSG3nO1 by the match of TSG3nCnTE with TSG3nCMP2E occurs when TSG3nCnTE = TSG3nDTC0 - 1.

Since TSG3nCMP0E + 1 is set to TSG3nCMP3E, TSG3nCnTE does not match TSG3nCMP3E. Therefore, TSG3nO1 output is fixed to active.

Setting of TSG3nO2 by the match of TSG3nCnTE with TSG3nCMP4E and clearing by the match with TSG3nCMP3E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO2 output is fixed to inactive.

When the value other than 0 is set to TSG3nHSPSHUE, set timing of TSG3nO1 is shifted to the right for the number of cycles set in TSG3nHSPSHUE. Note that the set timing is shifted only at operation start (TSG3nTE from 0 to 1) because TSG3nO1 output is fixed to active. For the operation when operation start, see **Section 25.4.7.7, (8) Timer Output Operation at Operation Start in HSP-PWM Mode.**

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

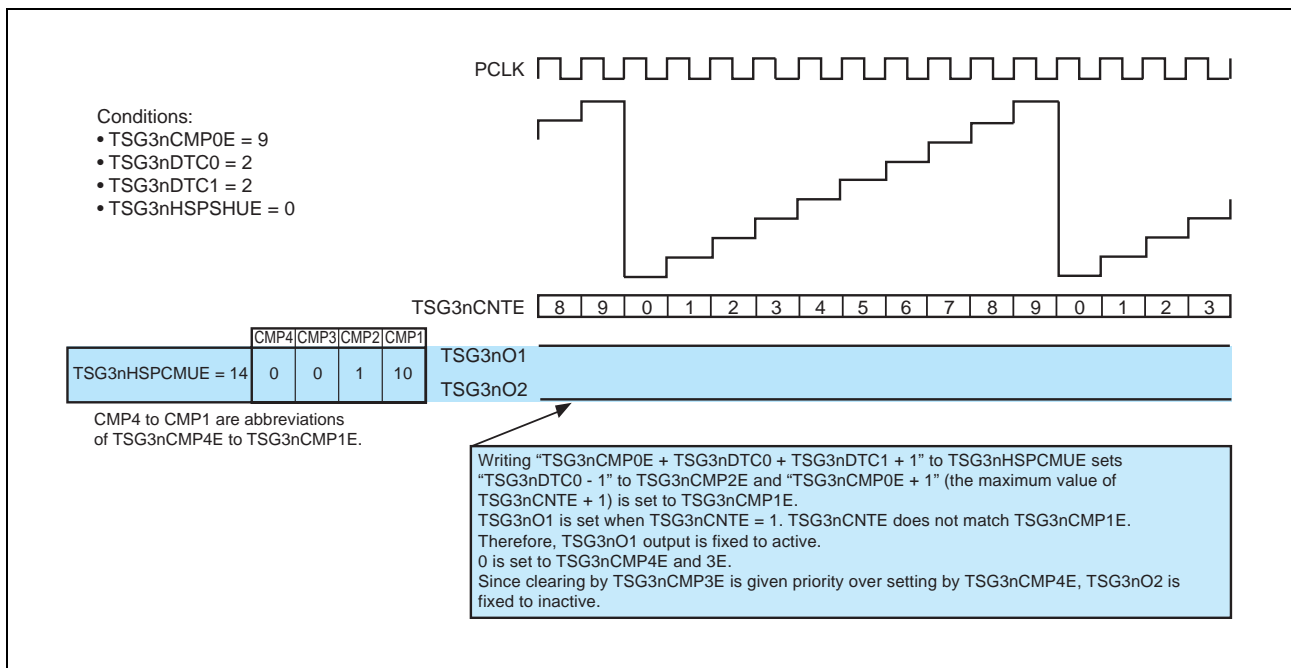


Figure 25.107 Waveform in HSP-PWM (TSG3nHSPCMUE is set to TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1)

NOTES

1. In HSP-PWM mode, TSG3nO1 and TSG3nO2 can be set at the same timing by setting the same value to TSG3nCMP4E and 2E. In this case, both TSG3nO1 and TSG3nO2 become inactive.
 2. In HSP-PWM mode, dead time counter keeps operating even when TSG is stopped (TSG3nTE = 0) and the dead time set in TSG3nO1 and TSG3nO2 are always inserted.
-

(7) PWM Adjustment at Reload Timing in HSP-PWM Mode

In HSP-PWM mode, PWM output width is adjusted when TSG3nHSPCMUE/VE/WE (PWM output width setting) is modified during operation. TSG3nO1-6 are set/cleared at reload timing and immediately switched to the output reflecting the new PWM output width.

TSG3nO1-6 are forcibly set/cleared according to the following formula. Even for adjustment, the dead time set with TSG3nDTC0 and 1 is always inserted to TSG3nO1-6. If values are directly written to TSG3nCMP1-12E, output adjustment is not performed at reload timing.

Formula for reload adjustment operation

Table 25.99 Formula for Reload Adjustment Operation
When TSG3nHSPSHUE/VE/WE is 0 (PWM shift width is set to 0)

| Pin | Set | Clear |
|-------------|------------------------------------|---------------------------------------|
| TSG3nO1/3/5 | $CMP0E + DTC1 + 1 < HSPCMUE/VE/WE$ | $HSPCMUE/VE/WE \leq CMP0E + DTC1 + 1$ |
| TSG3nO2/4/6 | $HSPCMUE/VE/WE = 0$ | $0 < HSPCMUE/VE/WE$ |

Table 25.100 Formula for Reload Adjustment Operation
When TSG3nHSPSHUE/VE/WE is not 0 (PWM shift width is not set to 0)

| Pin | Set | Clear |
|-------------|--|--|
| TSG3nO1/3/5 | (i) $CMP0E + DTC1 + 1 - HSPSHUE/VE/WE < HSPCMUE/VE/WE$ | $HSPCMUE/VE/WE \leq CMP0E + DTC1 + 1 - HSPSHUE/VE/WE$ |
| | (ii) $(CMP0E + 1) \times 2 + DTC1 - HSPSHUE/VE/WE < HSPCMUE/VE/WE$ | $HSPCMUE/VE/WE \leq (CMP0E + 1) \times 2 + DTC1 - HSPSHUE/VE/WE$ |
| TSG3nO2/4/6 | $HSPCMUE/VE/WE \leq CMP0E + 1 - HSPSHUE/VE/WE$ | $CMP0E + 1 - HSPSHUE/VE/WE < HSPCMUE/VE/WE$ |

If the value other than 0 is set as PWM shift width, set/clear condition for positive phase is determined whether the set shift width is greater than “CMP0E + DTC0 – 1” or not.

- (i) $HSPSHUE/VE/WE \leq CMP0 - DTC0 + 1$
- (ii) $HSPSHUE/VE/WE > CMP0 - DTC0 + 1$

“TSG3n” is omitted from the register names used in the calculation.

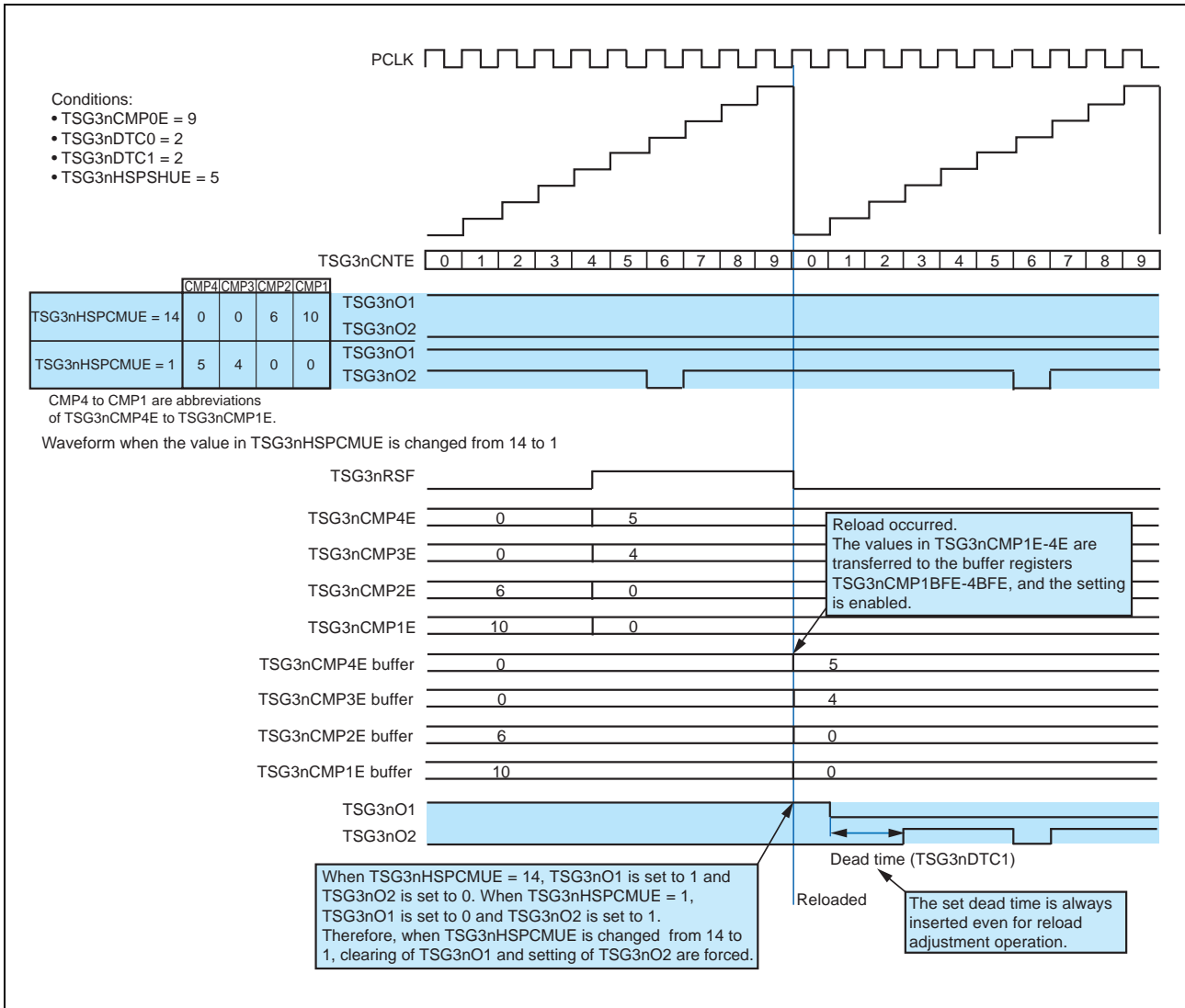


Figure 25.108 Detailed Timing Diagram of Reload Adjustment (Change of TSG3nHSPCMUE from 14 to 1)

(8) Timer Output Operation at Operation Start in HSP-PWM Mode

In HSP-PWM mode, TSG3nO1-6 are cleared when operation starts.

Then, TSG3nO1-6 are set/cleared as TSG3nCNTE counts up depending on the values set in TSG3nHSPCMUE/VE/WE (TSG3nCMP1E to 12E).

Even if TSG3nO1-6 are set before operation start and cleared at operation start, and then set again, the set dead time is always inserted.

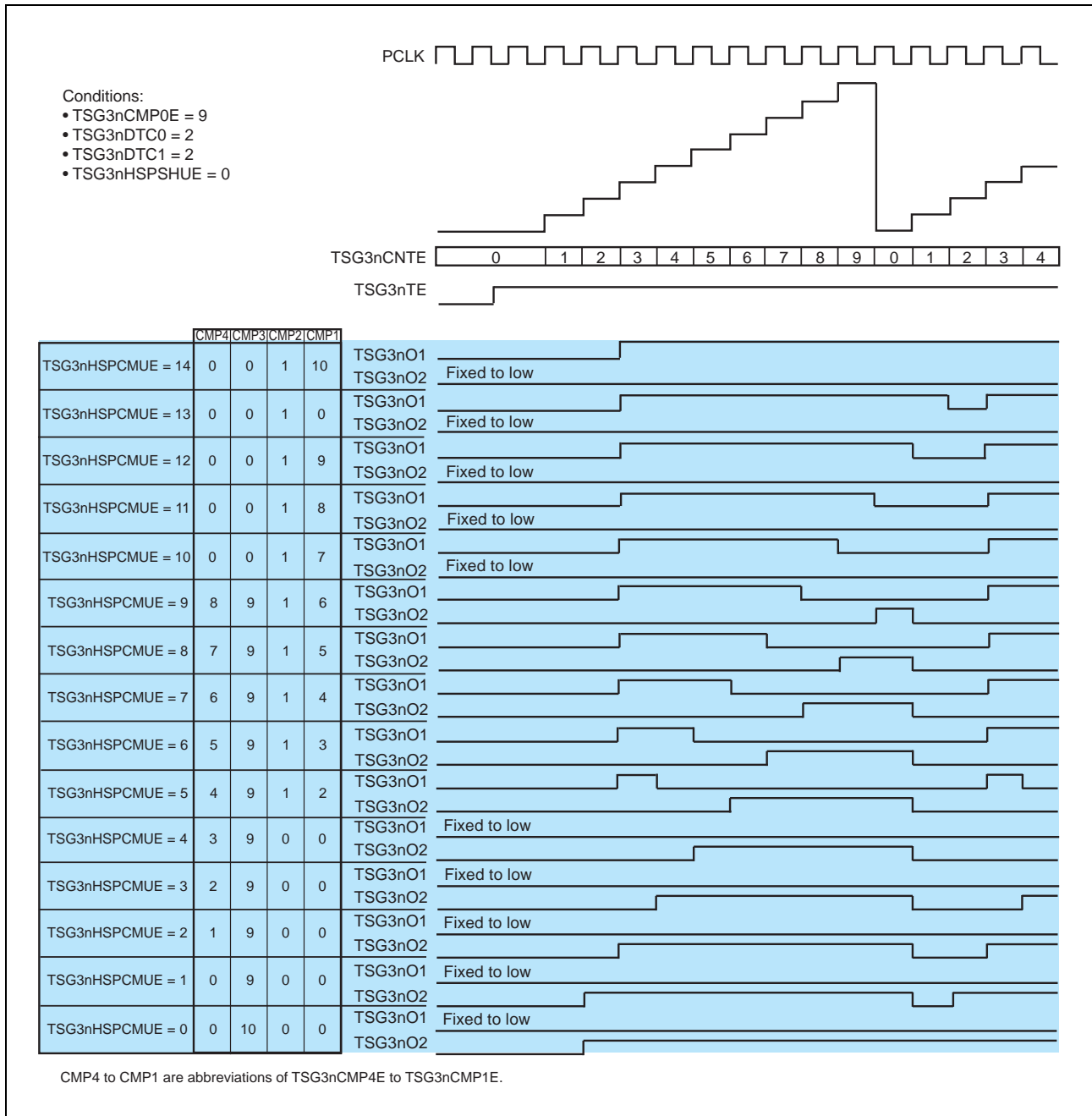


Figure 25.109 Timing Diagram of Operation Start in HSP-PWM Mode (TSG3nHSPSHUE = 0 (shift width is set to 0))

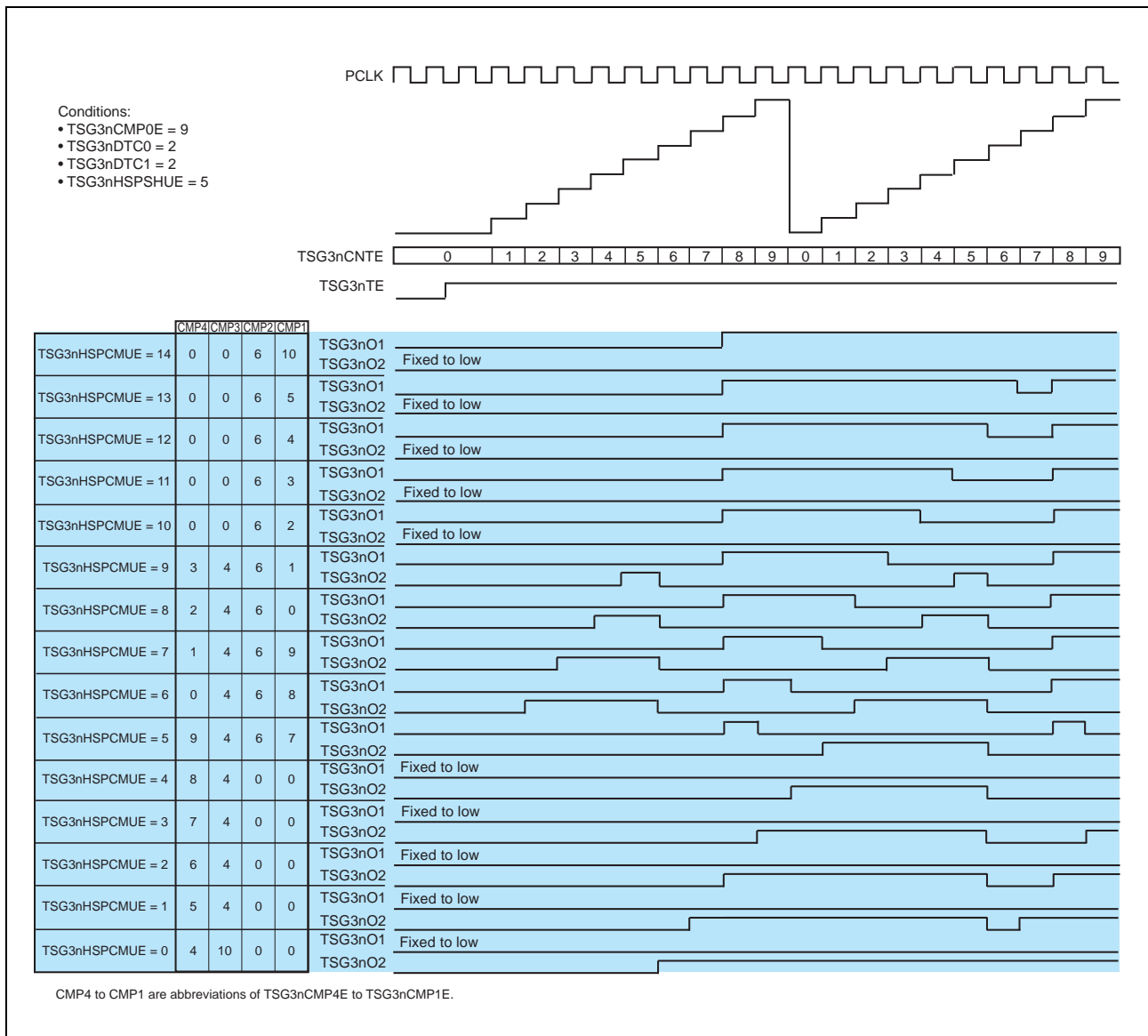


Figure 25.110 Timing Diagram of Operation Start in HSP-PWM Mode
(TSG3nHSPSHUE = 5 (shift width is set to five clock cycles))

25.4.7.8 Software Output Control Function

Software output control function is available in any mode except HSP-PWM mode. This function can switch between six output patterns for the TSG3nO1 to TSG3nO6 pins using TSG3nOPT0.TSG3nSOC, TSG3nIDC, and TSG3nOPT0.TSG3nSPC2-0.

When TSG3nSOC is switched from 0 to 1, the output control method of the TSG3nO1 to TSG3nO6 pins is switched to the software output control immediately. On the other hand, when TSG3nSOC is switched from 1 to 0, the software output control is released at the reload timing.

Table 25.101 Registers Associated with Software Output Control Function

| Register | Operation |
|-------------------------------|--|
| TSG3nOPT0.TSG3nSOC | TSG3nSOC = 1 |
| TSG3nOPT0.TSG3nSTE | TSG3nSTE = 0 |
| TSG3nOPT1.TSG3nSPC2-TSG3nSPC0 | Sets output patterns listed in the following Table 25.102 and Table 25.103 . |
| TSG3nOPT0.TSG3nIDC | Sets output pattern (electric current direction). |

Table 25.102 Output Patterns by Software Output Control (TSG3nOPT0.TSG3nIDC = 0)

TSG3nOPT0.TSG3nSOC = 1, TSG3nSTE = 0, TSG3nIDC = 0

| Output Pin | TSG3nSTR1.TSG3nOPF2-TSG3nOPF0 | | | | | | | |
|------------|-------------------------------|-------|-------|-------|-------|-------|-------|-------|
| | 101 | 100 | 110 | 010 | 011 | 001 | 000 | 111 |
| TSG3nO1 | ACT | ACT | ACT | INACT | INACT | INACT | INACT | ACT |
| TSG3nO2 | INACT | INACT | INACT | ACT | ACT | ACT | ACT | INACT |
| TSG3nO3 | INACT | INACT | ACT | ACT | ACT | INACT | INACT | ACT |
| TSG3nO4 | ACT | ACT | INACT | INACT | INACT | ACT | ACT | INACT |
| TSG3nO5 | ACT | INACT | INACT | INACT | ACT | ACT | INACT | ACT |
| TSG3nO6 | INACT | ACT | ACT | ACT | INACT | INACT | ACT | INACT |

Note: ACT: Indicates active level is output.
INACT: Indicates inactive level is output.

Table 25.103 Output Patterns by Software Output Control (TSG3nOPT0.TSG3nIDC = 1)

TSG3nOPT0.TSG3nSOC = 1, TSG3nSTE = 0, TSG3nIDC = 1

| Output Pin | TSG3nSTR1.TSG3nOPF2-TSG3nOPF0 | | | | | | | |
|------------|-------------------------------|-------|-------|-------|-------|-------|-------|-------|
| | 101 | 100 | 110 | 010 | 011 | 001 | 000 | 111 |
| TSG3nO1 | INACT | INACT | INACT | ACT | ACT | ACT | ACT | INACT |
| TSG3nO2 | ACT | ACT | ACT | INACT | INACT | INACT | INACT | ACT |
| TSG3nO3 | ACT | ACT | INACT | INACT | INACT | ACT | ACT | INACT |
| TSG3nO4 | INACT | INACT | ACT | ACT | ACT | INACT | INACT | ACT |
| TSG3nO5 | INACT | ACT | ACT | ACT | INACT | INACT | ACT | INACT |
| TSG3nO6 | ACT | INACT | INACT | INACT | ACT | ACT | INACT | ACT |

Note: ACT: Indicates active level is output.
INACT: Indicates inactive level is output.

Section 26 Timer Option (TAPA)

The Timer optional unit (TAPA) is Hi-Z output control for motor systems. TAPA is implemented 4 units.

This function forcibly sets the motor control output to the Hi-Z state, independently of control by the CPU.

26.1 Features of RH850/P1M-E TAPA

26.1.1 Number of Units

This microcontroller has the following number of TAPA units.

Table 26.1 Number of Units

| Product | RH850/P1M-E |
|-----------------|--------------------|
| Number of units | 4 |
| Name | TAPAn (n = 0 to 3) |

Table 26.2 Index

| Index | Meaning |
|-------|---|
| n | Throughout this section, the individual TAPA units are identified by the index "n" (n = 0 to 3), for example, TAPAnFLG for the TAPAn flag register. |

26.1.2 Register Base Address

TAPA register addresses are given as offsets from the given base address. Each of the TAPAn base addresses <TAPAn_base> is listed in the following table.

Table 26.3 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <TAPA0_base> | FFE9 0000 _H |
| <TAPA1_base> | FFE9 1000 _H |
| <TAPA2_base> | FFE9 2000 _H |
| <TAPA3_base> | FFE9 3000 _H |

26.1.3 Clock Supply

Clock supply by and to TAPA is listed in the following table.

Table 26.4 Clock Supply

| Unit Name | Clock for the Unit | Supply Clock Name |
|-----------|--------------------|-------------------------------------|
| TAPAn | PCLK | High-speed peripheral clock CLK_HSB |

26.1.4 Interrupt Requests

Interrupts or DMA/DTS triggers are not provided to the TAPA module.

26.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

26.1.6 Internal Signal

Internal signals of Timer Option are listed in the following table.

Table 26.5 TAPAn Internal Signals

| TAPAn Signal | Function | Connected to |
|----------------------------------|--|---|
| TAPA0 | | |
| TAPA0THASIN | Hi-Z control asynchronous input signal | PIC1 function |
| $\overline{\text{TAPA0THZOUT0}}$ | Hi-Z control output signal 0 (U phase) | Hi-Z control of TAUD0 U phase output signals (TAUD0O10, TAUD0O11) |
| $\overline{\text{TAPA0THZOUT1}}$ | Hi-Z control output signal 1 (V phase) | Hi-Z control of TAUD0 V phase output signals (TAUD0O12, TAUD0O13) |
| $\overline{\text{TAPA0THZOUT2}}$ | Hi-Z control output signal 2 (W phase) | Hi-Z control of TAUD0 W phase output signals (TAUD0O14, TAUD0O15) |
| TAPA1 | | |
| TAPA1THASIN | Hi-Z control asynchronous input signal | PIC1 function |
| $\overline{\text{TAPA1THZOUT0}}$ | Hi-Z control output signal 0 (U phase) | Hi-Z control of TAUD1 U phase output signals (TAUD1O10, TAUD1O11) |
| $\overline{\text{TAPA1THZOUT1}}$ | Hi-Z control output signal 1 (V phase) | Hi-Z control of TAUD1 V phase output signals (TAUD1O12, TAUD1O13) |
| $\overline{\text{TAPA1THZOUT2}}$ | Hi-Z control output signal 2 (W phase) | Hi-Z control of TAUD1 W phase output signals (TAUD1O14, TAUD1O15) |
| TAPA2 | | |
| TAPA2THASIN | Hi-Z control asynchronous input signal | PIC1 function |
| $\overline{\text{TAPA2THZOUT0}}$ | Hi-Z control output signal 0 | Hi-Z control of TSG30 output signals (TSG30O1-TSG30O6) |
| TAPA3 | | |
| TAPA3THASIN | Hi-Z control asynchronous input signal | PIC1 function |
| $\overline{\text{TAPA3THZOUT0}}$ | Hi-Z control output signal 0 | Hi-Z control of TSG31 output signals (TSG31O1-TSG31O6) |

26.1.7 Peripheral Configuration

The following figure shows peripheral configuration of TAPA.

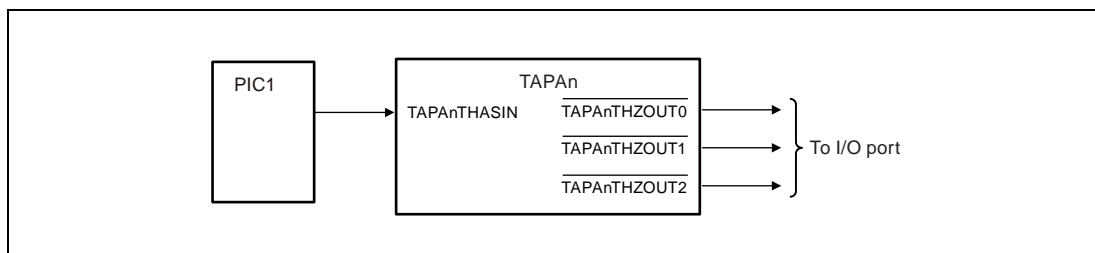


Figure 26.1 Peripheral Configuration of TAPA

The following describes peripheral configuration of TAPA.

- TAPAnTHASIN: Hi-Z control asynchronous input signal

This signal performs Hi-Z control by the source selected in PIC1.

For the sources selected in PIC1, see **Section 29.2.3.12, Hi-Z Control Function**.

26.2 Overview

26.2.1 Functional Overview

- Asynchronous Hi-Z control input signal (TAPAnTHASIN) enables individual asynchronous Hi-Z control of TAUD and TSG3 output.

26.2.2 Block Diagram

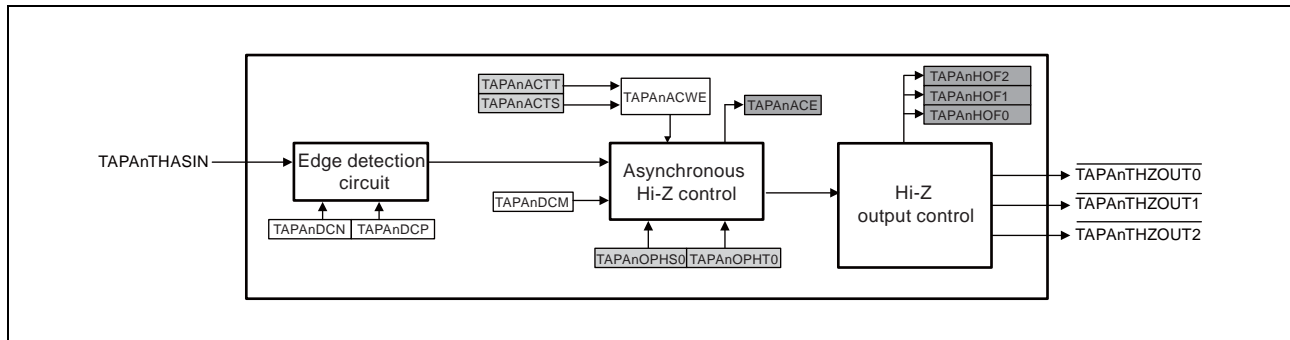


Figure 26.2 Block Diagram

26.3 Registers

26.3.1 List of Registers

TAPAn (n = 0 to 3) registers are listed in the following table.

<TAPAn_base> is defined in **Section 26.1.2, Register Base Address**.

| Module | Register Name | Symbol | Address |
|--------|---|-----------|--------------------------------|
| TAPAn | TAPAn control register 0 | TAPAnCTL0 | <TAPAn_base> + 20 _H |
| TAPAn | TAPAn flag register | TAPAnFLG | <TAPAn_base> + 00 _H |
| TAPAn | TAPAn asynchronous control write enable register | TAPAnACWE | <TAPAn_base> + 04 _H |
| TAPAn | TAPAn asynchronous control start trigger register | TAPAnACTS | <TAPAn_base> + 08 _H |
| TAPAn | TAPAn asynchronous control stop trigger register | TAPAnACTT | <TAPAn_base> + 0C _H |
| TAPAn | TAPAn Hi-Z start trigger register | TAPAnOPHS | <TAPAn_base> + 14 _H |
| TAPAn | TAPAn Hi-Z stop trigger register | TAPAnOPHT | <TAPAn_base> + 18 _H |

26.3.2 TAPAnCTL0 — TAPAn Control Register 0

Control register 0 is used to control Hi-Z.

A value in this register can only be rewritten in the following conditions.

- TAPAnFLG.TAPAnACE = 0 while TAPAn (n = 0, 1) and TAUDnTEm = 0 (m = 10 to 15) at the corresponding TAUDn master channel.
- TAPAnFLG.TAPAnACE = 0 while TAPAn (n = 2, 3).

Access This register can be read/written in 16-bit units.

Address <TAPAn_base> + 20_H

Value after reset 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|----------|----------|----------|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | TAPAnDCM | TAPAnDCN | TAPAnDCP | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R | R |

Table 26.6 TAPAnCTL0 Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|--------------------|--|----------|----------|-------------|---|---|-----------------------------|---|---|---|---|---|---|---|---|------------------------|
| 15 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 4 | TAPAnDCM | Clearing Condition Specification This control bit specifies the condition for clearing of the Hi-Z control outputs. 0: Manipulation of TAPAnOPHT0 is enabled regardless of the TAPAnTHASIN signal input level. 1: Manipulation of TAPAnOPHT0 is disabled when the TAPAnTHASIN input signal is at the active level. Manipulation of TAPAnOPHT0 is enabled when the TAPAnTHASIN signal input is inactive. | | | | | | | | | | | | | | | |
| 3, 2 | TAPAnDCN, TAPAnDCP | Hi-Z Input Edge Selection These control bits specify the valid edge of TAPAnTHASIN. <table border="1" data-bbox="678 1350 1420 1624"> <thead> <tr> <th>TAPAnDCN</th> <th>TAPAnDCP</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Does not detect valid edges</td> </tr> <tr> <td>0</td> <td>1</td> <td>Detects a rising edge as the valid edge (active level = high)</td> </tr> <tr> <td>1</td> <td>0</td> <td>Detects a falling edge as the valid edge (active level = low)</td> </tr> <tr> <td>1</td> <td>1</td> <td>Setting is prohibited.</td> </tr> </tbody> </table> | TAPAnDCN | TAPAnDCP | Description | 0 | 0 | Does not detect valid edges | 0 | 1 | Detects a rising edge as the valid edge (active level = high) | 1 | 0 | Detects a falling edge as the valid edge (active level = low) | 1 | 1 | Setting is prohibited. |
| TAPAnDCN | TAPAnDCP | Description | | | | | | | | | | | | | | | |
| 0 | 0 | Does not detect valid edges | | | | | | | | | | | | | | | |
| 0 | 1 | Detects a rising edge as the valid edge (active level = high) | | | | | | | | | | | | | | | |
| 1 | 0 | Detects a falling edge as the valid edge (active level = low) | | | | | | | | | | | | | | | |
| 1 | 1 | Setting is prohibited. | | | | | | | | | | | | | | | |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |

26.3.3 TAPAnFLG — TAPAn Flag Register

This flag register is used to control Hi-Z.

Access This register can only be read in 16-bit units.

Address <TAPAn_base> + 00_H

Value after reset 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|---------------|---------------|---------------|---|---|---|---|---|---|---|--------------|
| | — | — | — | — | — | TAPAnH OF2 | TAPAnH OF1 | TAPAnH OF0 | — | — | — | — | — | — | — | TAPAnA CE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 26.7 TAPAnFLG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|---|
| 15 to 11 | Reserved | When read, the value after reset is read. |
| 10 to 8 | TAPAnHOF _m | <p>HZOUT_m output monitor bit (<i>m</i> = 0 to 2)</p> <p>This bit monitors the $\overline{\text{TAPAnTHZOUT}}_m$ output.</p> <p>0: The $\overline{\text{TAPAnTHZOUT}}_m$ output is at the high level.</p> <p>1: The $\overline{\text{TAPAnTHZOUT}}_m$ output is at the low level.*1</p> |
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | TAPAnACE | <p>Asynchronous Hi-Z Control Enable</p> <p>This bit indicates the state of asynchronous Hi-Z control (TAPAnTHASIN).</p> <p>0: Asynchronous Hi-Z control is disabled.</p> <p>1: Asynchronous Hi-Z control is enabled.</p> <p>The conditions for setting and clearing this bit are as follows.</p> <p>Clearing condition: Writing 1 to TAPAnACTT while TAPAnACWE = 1.</p> <p>Setting condition: Writing 1 to TAPAnACTS while TAPAnACWE = 1.</p> |

Note 1. TAPAnHOF_m (*m* = 1, 2) is enabled when TAPAn (*n* = 0, 1).

26.3.4 TAPAnACWE — TAPAn Asynchronous Control Write Enable Register

This register enables writing for asynchronous Hi-Z control.

Access This register can be read/written in 8-bit units.

Address <TAPAn_base> + 04_H

Value after reset 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------|
| | — | — | — | — | — | — | — | TAPAnACWE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 26.8 TAPAnACWE Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TAPAnACWE | Asynchronous Control Write Enable This is a write-enable bit for asynchronous Hi-Z control. After 1 has been written to this bit, it is automatically cleared to 0 by writing 1 to TAPAnACTS or TAPAnACTT. 0: Disables writing to TAPAnACTS and TAPAnACTT. 1: Enables writing to TAPAnACTS and TAPAnACTT. |

26.3.5 TAPAnACTS — TAPAn Asynchronous Control Start Trigger Register

This register enables the start trigger for asynchronous Hi-Z control.

Access This register can only be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + 08_H

Value after reset 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------|
| | — | — | — | — | — | — | — | TAPAnACTS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 26.9 TAPAnACTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TAPAnACTS | Asynchronous Control Start Trigger This bit enables the start trigger for asynchronous Hi-Z control. The setting of this bit is only valid when TAPAnACWE = 1. 0: Writing 0 to this bit is ignored (no function). 1: Enables asynchronous Hi-Z control if TAPAnACWE = 1. |

26.3.6 TAPAnACTT — TAPAn Asynchronous Control Stop Trigger Register

This register enables the stop trigger for asynchronous Hi-Z control.

Access This register can only be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + 0C_H

Value after reset 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------|
| | — | — | — | — | — | — | — | TAPAnACTT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 26.10 TAPAnACTT Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TAPAnACTT | Asynchronous Control Stop Trigger This bit enables the stop trigger for asynchronous Hi-Z control. The setting of this bit is only valid when TAPAnACWE = 1. 0: Writing 0 to this bit is ignored (no function). 1: Stops asynchronous Hi-Z control if TAPAnACWE = 1. |

26.3.7 TAPAnOPHS — TAPAn Hi-Z Start Trigger Register

This register sets the start trigger for a Hi-Z control signal ($\overline{\text{TAPAnTHZOUTm}}$) ($m = 0$ to 2^{*1})

Note 1. TAPAn ($n = 2, 3$) are not available when $m = 1, 2$.

Access This register can only be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + 14_H

Value after reset 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TAPAnOPHS0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 26.11 TAPAnOPHS Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TAPAnOPHS0 | Hi-Z Control Signal Start Trigger 0 This bit sets the start trigger for a Hi-Z control signal. 0: The read value is always 0 and writing 0 to this bit is ignored. 1: Sets the Hi-Z control signal ($\overline{\text{TAPAnTHZOUTm}}$) ($m = 0$ to 2^{*1}) to the low level. |

26.3.8 TAPAnOPHT — TAPAn Hi-Z Stop Trigger Register

This register sets the stop trigger for a Hi-Z signal ($\overline{\text{TAPAnTHZOUTm}}$) ($m = 0$ to 2^{*1})

Note 1. TAPAn ($n = 2, 3$) are not available when $m = 1, 2$.

Access This register can only be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + 18_H

Value after reset 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TAPAnOPHT0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 26.12 TAPAnOPHT Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TAPAnOPHT0 | Hi-Z Control Signal Stop Trigger 0 Sets the stop trigger for a Hi-Z control signal. 0: The read value is always 0 and writing 0 to this bit is ignored. 1: Sets Hi-Z control signal ($\overline{\text{TAPAnTHZOUTm}}$) ($m = 0$ to 2^{*1}) to the high level. |

26.4 Function

26.4.1 Asynchronous Hi-Z Control Function

Abnormal operation in timer motor-control under CPU control leads to rotation of the externally connected motor also becoming abnormal. In such cases, this function forcibly sets the motor control output to the Hi-Z state, independently of control by the CPU.

26.4.1.1 Overview

The following method is available for controlling Hi-Z.

- Asynchronous input Hi-Z control by Hi-Z control asynchronous input signal (TAPAnTHASIN)
 - Controls the Hi-Z control output signals $\overline{\text{TAPAnTHZOUT0}}$ (U phase), $\overline{\text{TAPAnTHZOUT1}}$ (V phase), $\overline{\text{TAPAnTHZOUT2}}$ (W phase) asynchronously at TAPAn (n = 0, 1).
 - Controls the Hi-Z control output signals $\overline{\text{TAPAnTHZOUT0}}$ asynchronously at TAPAn (n = 2, 3).

Asynchronous Hi-Z Control and Its Operation

| Function | Operation |
|--|--|
| Asynchronous Hi-Z control corresponding to the pin input | This function detects asynchronous pin inputs and forcibly stops TOUTn output from the corresponding timer module (TAUD, TSG3) in response. Device port outputs become Hi-Z while TAPAnTHASIN is active and until software sends a stop request. |

26.4.1.2 Example of System Configuration

When an external error detection signal is received, an interrupt is generated and, at the same time, the motor-driving signal output is set to the Hi-Z state.

This module assumes that microcontroller operation may hang when an error occurs. To handle such situations, external error detection signals are continuously processed so that the motor-driving signal can be set to the Hi-Z state even if no clock signal is being supplied.

This module only detects an error when the edge of an error-detection signal is detected. A fixed output level is not detected as an error (the signal has no edge.)

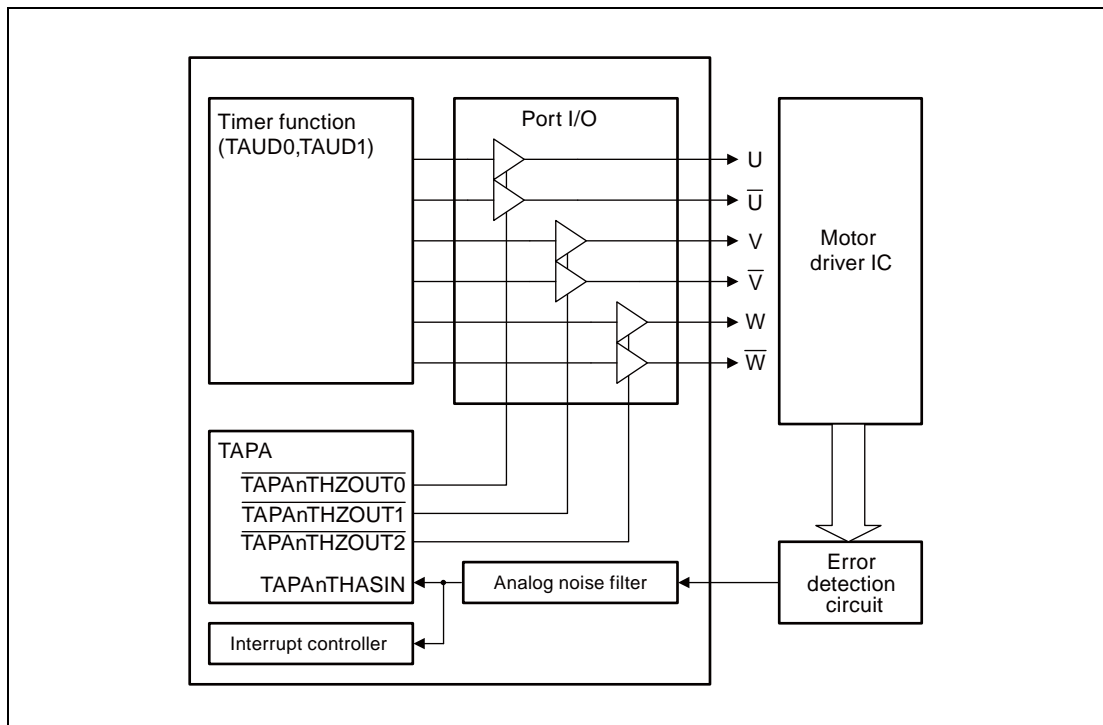


Figure 26.3 Example of System Configuration of Asynchronous Hi-Z Control for Pin Inputs (TAUD0, TAUD1)

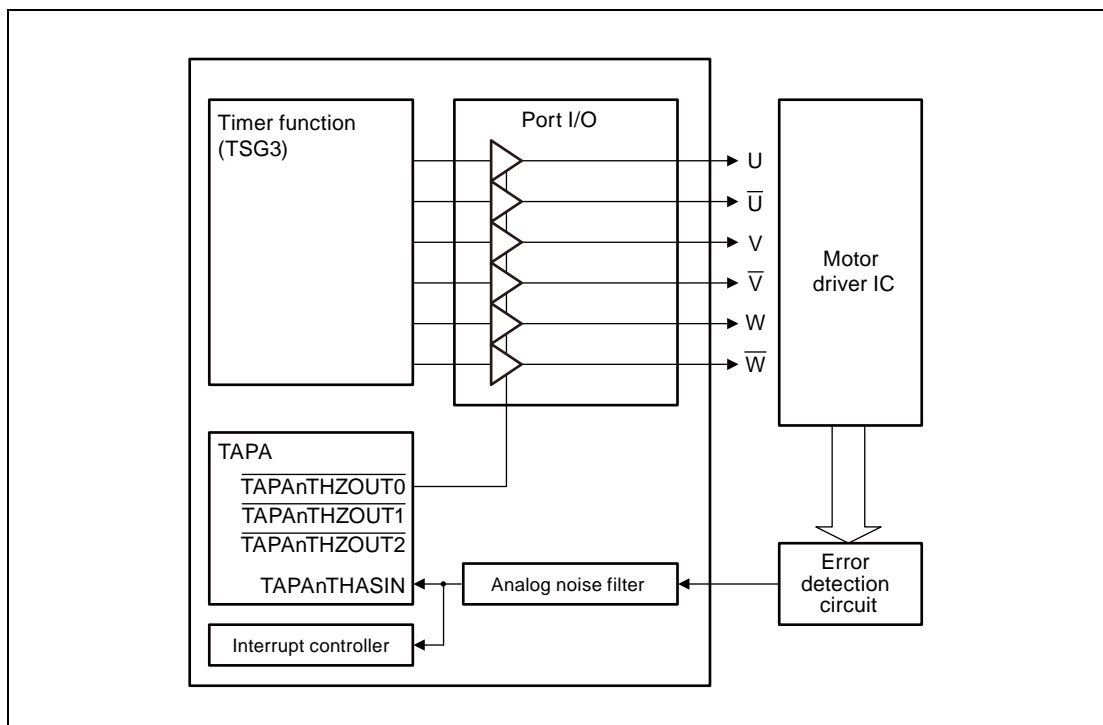


Figure 26.4 Example of System Configuration of Asynchronous Hi-Z Control for Pin Inputs (TSG30, TSG31)

26.4.1.3 Basic Operation

Setting examples are described as follows.

When $TAPAnCTL0.TAPAnDCM = 0$, $TAPAnDCP = 1$, and $TAPAnDCN = 0$

$\overline{TAPAnTHZOUT0}$ goes to the low level on detection of a valid edge of the asynchronous input ($TAPAnTHASIN$).

Output is forcibly stopped (by port control for Hi-Z output) as long as the $\overline{TAPAnTHZOUT0}$ output is at the low level.

$\overline{TAPAnTHZOUT0}$ goes to the high level in response to writing 1 to the bit 0 of TAPAn Hi-Z stop trigger ($TAPAnOPHT0$), regardless of the level of $TAPAnTHASIN$.

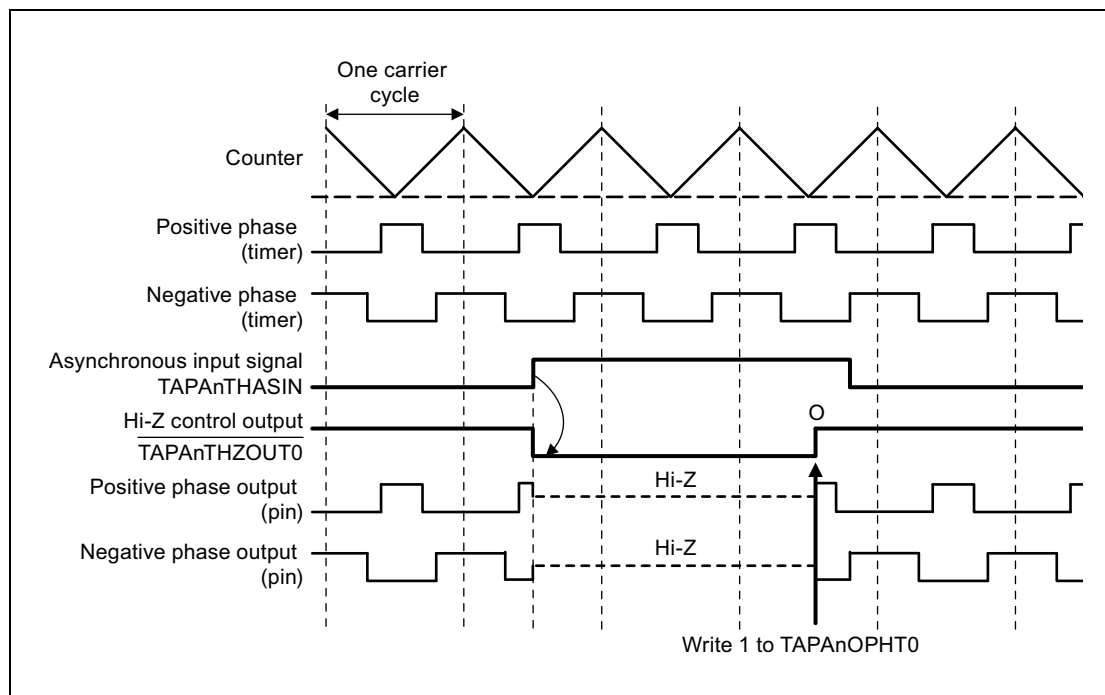


Figure 26.5 $\overline{TAPAnTHZOUT0}$ Operation when $TAPAnDCM = 0$, $TAPAnDCP = 1$, and $TAPAnDCN = 0$

When TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0

$\overline{\text{TAPAnTHZOUT0}}$ goes to the low level in response to detection of a valid edge of the asynchronous input signal (TAPAnTHASIN).

Output is forcibly stopped (by port control for Hi-Z output) as long as the $\overline{\text{TAPAnTHZOUT0}}$ output is at the low level.

Writing of 1 to Hi-Z stop trigger 0 (TAPAnOPHT0) is ignored as long as the asynchronous input signal (TAPAnTHASIN) is at the active level (high because TAPAnDCP = 1).

After the asynchronous input signal (TAPAnTHASIN) is switched to the inactive level (low because TAPAnDCP = 1), $\overline{\text{TAPAnTHZOUT0}}$ goes to the high level when 1 is written to Hi-Z stop trigger 0 (TAPAnOPHT0).

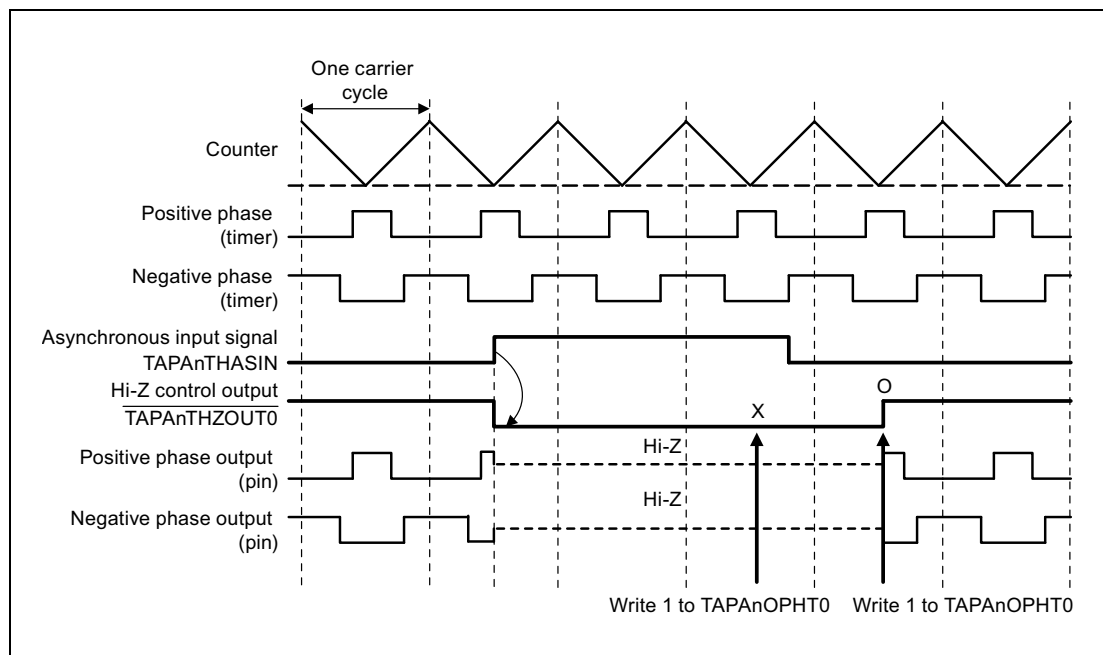


Figure 26.6 $\overline{\text{TAPAnTHZOUT0}}$ Operation when TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0

26.4.1.4 Asynchronous Hi-Z Control by Software Trigger

This module allows software control of the output of Hi-Z control signals. Hi-Z start trigger 0 (TAPAnOPHS0) and Hi-Z stop trigger 0 (TAPAnOPHT0) are used to control TAPAnTHZOUT0, TAPAnTHZOUT1*1, and TAPAnTHZOUT2*1.

Note 1. Not available when TAPAn (n = 2, 3).

Operation of the Hi-Z Start Trigger (TAPAnOPHS)

| TAPAnDCM | Operation |
|----------|---|
| 0/1 | Writing 1 to the TAPAnOPHS0 bit places the TAPAnTHZOUT0, TAPAnTHZOUT1, and TAPAnTHZOUT2 at the low level. |

Operation of the Hi-Z Stop Trigger (TAPAnOPHT) during Asynchronous Input Hi-Z Control

The Hi-Z stop trigger is enabled in the following conditions.

| TAPAnDCM | Operation |
|----------|---|
| 0 | Writing 1 to the TAPAnOPHT0 bit places the TAPAnTHZOUT0, TAPAnTHZOUT1, and TAPAnTHZOUT2 at the high level. |
| 1 | If TAPAnTHASIN is at the inactive level, writing 1 to the TAPAnOPHT0 bit places the TAPAnTHZOUT0, TAPAnTHZOUT1, and TAPAnTHZOUT2 at the high level. If TAPAnTHASIN is at the active level, writing of 1 to the TAPAnOPHT0 bit is ignored. |

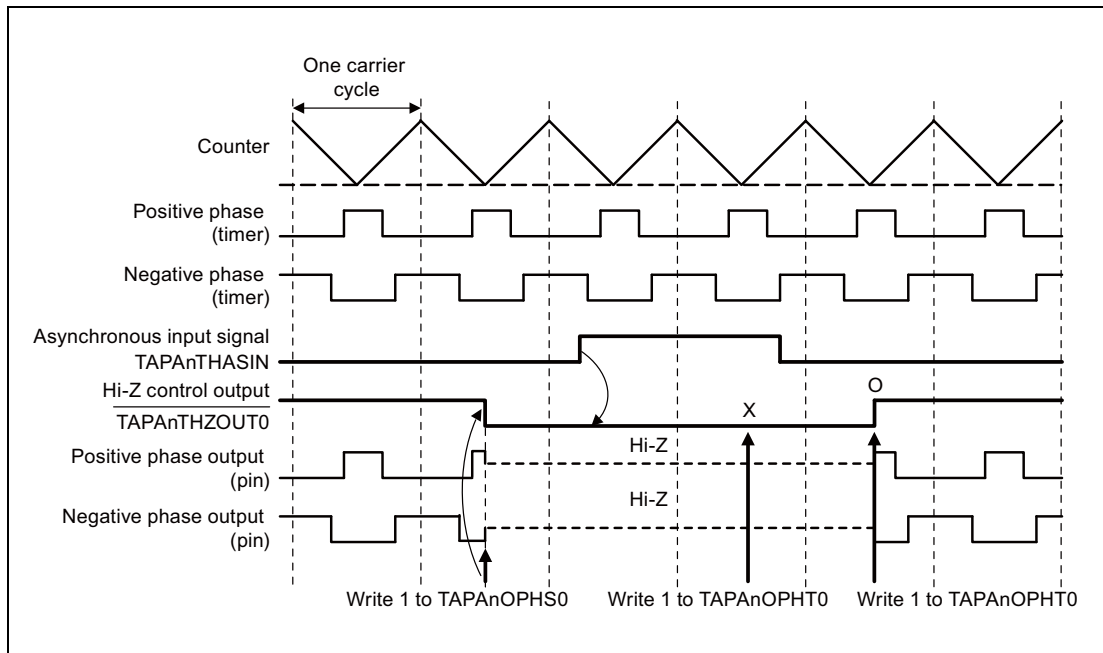


Figure 26.7 TAPAnTHZOUT0 Operation TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0

26.4.1.5 Operating Procedure

An example of the operating procedure for Hi-Z control in response to asynchronous input is as follows (the table only covers settings for the timer option module because this operation does not depend on timer operations).

| | Operation | State of TAPA |
|--------------------|---|--|
| Initial settings | Setting in the TAPAnCTL0 register. Set TAPAnDCP and TAPAnDCN (input edge selection). Set TAPAnDCM (clearing mode selection). | Hi-Z control in response to asynchronous input is stopped (TAPAnFLG.TAPAnACE = 0). |
| Starting operation | Setting in the TAPAnACWE register: Set the TAPAnACWE bit to 1. Setting in the TAPAnACTS register: Set the TAPAnACTS bit to 1. | Writing to TAPAnACTS is enabled. TAPAnFLG.TAPAnACE = 1, enabling Hi-Z control in response to asynchronous input. |
| During operation | To start Hi-Z control of an output from the timer: <ul style="list-style-type: none"> – Controlled by the TAPAnOPHS0 in TAPA. – Controlled by the Hi-Z control asynchronous input signal (TAPAnTHASIN). To stop Hi-Z control of output from the timer: <ul style="list-style-type: none"> – Controlled by the TAPAnOPHT0 in TAPA. (TAPAnDCM = 0) – Controlled by the TAPAnOPHT0 when Hi-Z input signal in TAPA (TAPAnTHASIN) is inactive level. The state of TAPA operations can be read from the TAPAnFLG register at all times. | On detection of input of the starting edge of the Hi-Z input signal (TAPAnTHASIN) or setting of the start trigger bit (TAPAnOPHS0 = 1), the Hi-Z controller switches the <u>TAPAnTHZOUT0</u> , <u>TAPAnTHZOUT1*1</u> , and <u>TAPAnTHZOUT2*1</u> pins to low-level output. In accord with the operating mode settings in TAPAnDCM, the Hi-Z controller switches the <u>TAPAnTHZOUT0</u> , <u>TAPAnTHZOUT1*1</u> , and <u>TAPAnTHZOUT2*1</u> pins to high-level outputs in response to setting of the stop trigger bit (TAPAnOPHT0 = 1). |
| Stopping operation | Setting in the TAPAnACWE register: Set the TAPAnACWE bit to 1. Setting in the TAPAnACTT register: Set the TAPAnACTT bit to 1. | Writing to TAPAnACTT is enabled. TAPAnFLG.TAPAnACE = 0, stopping asynchronous Hi-Z control |

Note 1. Not available when TAPAn (n = 2, 3).

Section 27 Timer Pattern Buffer (TPBA)

The Timer Pattern Buffer (TPBA) is 16-bit PWM timer with the duty setting buffer. TPBA is implemented 2 units.

27.1 Features of RH850/P1M-E TPBA

27.1.1 Units and Channels

This microcontroller has the following number of TPBA units.

Each TPBA unit has one channel interface.

Table 27.1 Number of Units

| Product Name | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|------------------------------|------------------------------|
| Number of units | 2 | 2 |
| Name | TPBA _n (n = 0, 1) | TPBA _n (n = 0, 1) |

Table 27.2 Index

| Index | Meaning |
|-------|--|
| n | Throughout this section, the individual TPBA units are identified by the index “n” (n = 0, 1); for example, TPBA _n CTL is the TPBA _n control register. |
| m | The number of buffers are indicated by the index “m” (m = 00 to 63). |

27.1.2 Register Base Address

TPBA base addresses are listed in the following table.

TPBA register addresses are given as offsets from the base addresses in general.

Table 27.3 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <TPBA0_base> | FFEA 0000 _H |
| <TPBA1_base> | FFEA 1000 _H |

27.1.3 Clock Supply

Clock supply by and to TPBA is listed in the following table.

Table 27.4 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-------------------|-----------------|-------------------------------------|
| TPBA _n | PCLK | High-speed peripheral clock CLK_HSB |

27.1.4 Interrupt Requests

TPBA interrupt requests are listed in the following table.

Table 27.5 Interrupt Requests

| Unit Interrupt Name | Outline | Interrupt Number | DMA/DTS Trigger Number |
|---------------------|---|------------------|------------------------|
| TPBA0 | | | |
| INTTPBA0IPRD | TPBA0 Period match detection interrupt | 286 | 111 |
| INTTPBA0IDTY | TPBA0 Duty match detection interrupt | 287 | 112 |
| INTTPBA0IPAT | TPBA0 Pattern number matching detection interrupt | 288 | 113 |
| TPBA1 | | | |
| INTTPBA1IPRD | TPBA1 Period match detection interrupt | 289 | 114 |
| INTTPBA1IDTY | TPBA1 Duty match detection interrupt | 290 | 115 |
| INTTPBA1IPAT | TPBA1 Pattern number matching detection interrupt | 291 | 116 |

27.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

27.1.6 External Input/Output Signals

External input/output signals of TPBA are listed in the following table.

Table 27.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|------------------|-----|--------------|----------------------------------|
| TPBA0 | | | |
| TPBA0O | O | Timer output | TPBA0O |
| TPBA1 | | | |
| TPBA1O | O | Timer output | TPBA1O |

27.2 Overview

27.2.1 Functional Overview

TPBA_n is a 16-bit PWM timer with the duty setting buffer.

- Count clock resolution: Min. 12.5 ns (count clock: 80 MHz)
- 16-bit counter
- 16-bit duty register
- 16-bit period setting register
- 7-bit address counter register
- 7-bit pattern number setting register
- Interrupt request signals
 - Period-matched detection interrupt
 - Duty-cycle-matched detection interrupt
 - Number-of-patterns matched detection interrupt
- Number of duty patterns
 - 64 patterns (16 bits) or 128 patterns (8 bits)
- Automatic duty generation according to the number of patterns
- Output control by software
- The count clock can be selected from PCLK, PCLK/2, PCLK/4, and PCLK/8 according to the prescaler set value.
- Synchronous start with another timer

27.2.2 Block Diagram

The following block diagram shows the main components of TPBA.

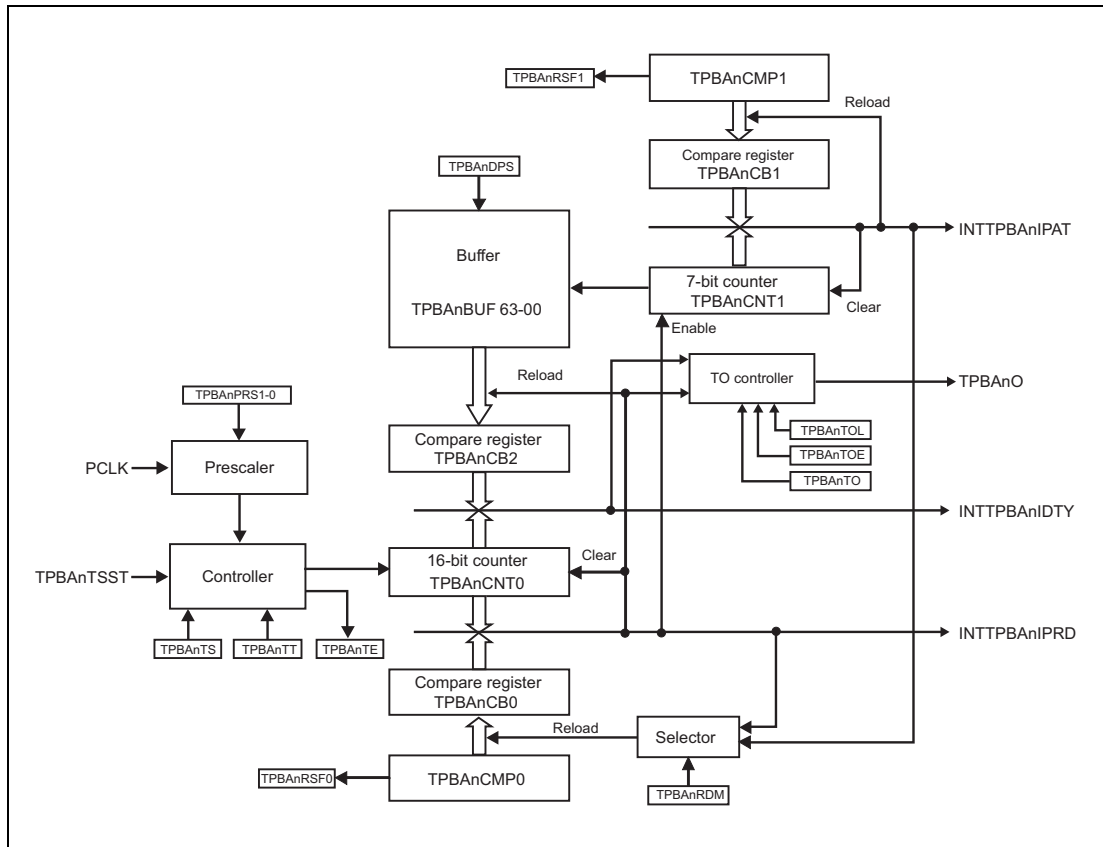


Figure 27.1 Block Diagram of TPBA

- TPBA_nTSST Simultaneous start trigger (input by PIC function)

27.3 Registers

27.3.1 List of Registers

TPBA registers are listed in the following table.

For information on <TPBA_base>, see **Section 27.1.2, Register Base Address**.

Table 27.7 List of Registers

| Module Name | Register Name | Symbol | Address |
|-------------------|---|------------------------------------|--|
| TPBA _n | TPBA _n control register | TPBA _n CTL | <TPBA _n _base> + 200 _H |
| TPBA _n | TPBA _n reload data mode register | TPBA _n RDM | <TPBA _n _base> + 118 _H |
| TPBA _n | TPBA _n reload status register | TPBA _n RSF | <TPBA _n _base> + 110 _H |
| TPBA _n | TPBA _n reload data trigger register | TPBA _n RDT | <TPBA _n _base> + 114 _H |
| TPBA _n | TPBA _n timer output enable register | TPBA _n TOE | <TPBA _n _base> + 120 _H |
| TPBA _n | TPBA _n timer output register | TPBA _n TO | <TPBA _n _base> + 11C _H |
| TPBA _n | TPBA _n timer output level register | TPBA _n TOL | <TPBA _n _base> + 124 _H |
| TPBA _n | TPBA _n period setting register | TPBA _n CMP0 | <TPBA _n _base> + 100 _H |
| TPBA _n | TPBA _n duty setting register | TPBA _n BUF _m | <TPBA _n _base> + m × 4 _H |
| TPBA _n | TPBA _n pattern number setting register | TPBA _n CMP1 | <TPBA _n _base> + 104 _H |
| TPBA _n | TPBA _n timer counter register | TPBA _n CNT0 | <TPBA _n _base> + 108 _H |
| TPBA _n | TPBA _n address counter register | TPBA _n CNT1 | <TPBA _n _base> + 10C _H |
| TPBA _n | TPBA _n enable status register | TPBA _n TE | <TPBA _n _base> + 128 _H |
| TPBA _n | TPBA _n start trigger register | TPBA _n TS | <TPBA _n _base> + 12C _H |
| TPBA _n | TPBA _n stop trigger register | TPBA _n TT | <TPBA _n _base> + 130 _H |

27.3.2 TPBA_nCTL — TPBA_n Control Register

This register specifies the operation of the TPBA_n.

Access: This register can be read/written in 8-bit units.

Address: <TPBA_n_base> + 200_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|----------------------------|-----|---|---|---|-----------------------|
| | — | — | TPBA _n PRS[1:0] | | — | — | — | TPBA _n DPS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R | R/W |

Table 27.8 TPBA_nCTL Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|------------------------|-----------------------------|--|------------------------|------------------------|-------------|---|---|-------------------|---|---|---------------------|---|---|---------------------|---|---|---------------------|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 5, 4 | TPBA _n PRS [1:0] | Selects the count clock. <table border="1" data-bbox="678 891 1417 1081"> <thead> <tr> <th>TPBA_nPRS1</th> <th>TPBA_nPRS0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>PCLK is selected.</td> </tr> <tr> <td>0</td> <td>1</td> <td>PCLK/2 is selected.</td> </tr> <tr> <td>1</td> <td>0</td> <td>PCLK/4 is selected.</td> </tr> <tr> <td>1</td> <td>1</td> <td>PCLK/8 is selected.</td> </tr> </tbody> </table> | TPBA _n PRS1 | TPBA _n PRS0 | Description | 0 | 0 | PCLK is selected. | 0 | 1 | PCLK/2 is selected. | 1 | 0 | PCLK/4 is selected. | 1 | 1 | PCLK/8 is selected. |
| TPBA _n PRS1 | TPBA _n PRS0 | Description | | | | | | | | | | | | | | | |
| 0 | 0 | PCLK is selected. | | | | | | | | | | | | | | | |
| 0 | 1 | PCLK/2 is selected. | | | | | | | | | | | | | | | |
| 1 | 0 | PCLK/4 is selected. | | | | | | | | | | | | | | | |
| 1 | 1 | PCLK/8 is selected. | | | | | | | | | | | | | | | |
| 3 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 0 | TPBA _n DPS | Selects the duty setting pattern mode. 0: 16 bits × 64 patterns mode 1: 8 bits × 128 patterns mode | | | | | | | | | | | | | | | |

CAUTION

This register should be set when the timer is stopped (TPBA_nTE = 0). If this register is erroneously rewritten, set the register again after stopping the timer.

27.3.3 TPBAnRDM — TPBAn Reload Data Mode Register

This register controls the reload timing of the TPBAn period setting register and TPBAn timer output level register values.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 118_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------|
| | — | — | — | — | — | — | — | TPBAnRDM0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 27.9 TPBAnRDM Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TPBAnRDM0 | Controls the reload timing of the TPBAn period setting register (TPBAnCMP0) and TPBAn timer output level register (TPBAnTOL) values. 0: Reloads the values synchronously with a number-of-patterns matched detection interrupt (INTTPBAnIPAT). 1: Reloads the values synchronously with a period-matched detection interrupt (INTTPBAnIPRD). |

CAUTION

Although this register can be rewritten during operation, the rewritten value is reflected immediately. Accordingly, during operation, this register should be rewritten when the reload request flag (TPBAnRSF) is 0.

27.3.4 TPBAnRSF — TPBAn Reload Status Register

This register indicates whether or not reload requests from the corresponding registers have been generated.

Access: This register can be read in 8-bit units.

Address: <TPBAn_base> + 110_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|-----------|-----------|
| | — | — | — | — | — | — | TPBAnRSF1 | TPBAnRSF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 27.10 TPBAnRSF Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. |
| 1 | TPBAnRSF1 | Indicates whether or not a reload request from TPBAnCMP1 has been generated. 0: No reload request generated or reload completed. 1: Reload request has been generated. This bit is set to 1 when 1 is written to the TPBAnRDT1 bit in the TPBAnRDT register. This bit is cleared at the timing when reload is performed. |
| 0 | TPBAnRSF0 | Indicates whether or not a reload request from TPBAnCMP0 and TPBAnTOL has been generated. 0: No reload request generated or reload completed. 1: Reload request has been generated. This bit is set to 1 when 1 is written to the TPBAnRDT0 bit in the TPBAnRDT register. This bit is cleared at the timing when reload is performed. |

27.3.5 TPBAnRDT — TPBAn Reload Data Trigger Register

This register enables reload of the register values.

Access: This register can be written in 8-bit units. It is always read as 0.

Address: <TPBAn_base> + 114_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|-----------|-----------|
| | — | — | — | — | — | — | TPBAnRDT1 | TPBAnRDT0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | W | W |

Table 27.11 TPBAnRDT Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 2 | Reserved | When writing, write the value after reset. |
| 1 | TPBAnRDT1 | Enables reload of the TPBAnCMP1 values. 0: Write access is ignored. 1: Reload is enabled (TPBAnRSF1 is set to 1). The values are updated simultaneously at the next reload timing (reload). |
| 0 | TPBAnRDT0 | Enables reload of the TPBAnCMP0 and TPBAnTOL values. 0: Write access is ignored. 1: Reload is enabled (TPBAnRSF0 is set to 1). The values are updated simultaneously at the next reload timing (reload). |

27.3.6 TPBAnTOE — TPBAn Timer Output Enable Register

This register enables or disables the timer output.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 120_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------|
| | — | — | — | — | — | — | — | TPBAnTOE0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 27.12 TPBAnTOE Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TPBAnTOE0 | Enables or disables the timer output (TPBAnO). 0: Disables the timer output based on counter operation. 1: Enables the timer output based on counter operation. <ul style="list-style-type: none"> When the timer output is disabled, the level specified in TPBAnTO is output from the TPBAnO pin, and can be controlled by software. When the timer output is enabled, TPBAnTO is set or cleared by the timer operation, and a PWM signal is output. Write access is prohibited (ignored). |

27.3.7 TPBA_nTO — TPBA_n Timer Output Register

This register controls or reads timer output level.

Access: This register can be read/written in 8-bit units.

Address: <TPBA_n_base> + 11C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------------------|
| | — | — | — | — | — | — | — | TPBA _n TO0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 27.13 TPBA_nTO Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TPBA _n TO0 | Sets or indicates the output level of TPBA _n O pin <ul style="list-style-type: none"> When the timer output is disabled (TPBA_nTOE.TPBA_nTOE0 = 0) <ul style="list-style-type: none"> 0: Outputs low level. 1: Outputs high level. The output level can be controlled by rewriting this register while the timer output is disabled. When the timer output is enabled (TPBA_nTOE.TPBA_nTOE0 = 1) <ul style="list-style-type: none"> 0: Low level is being output by the timer output. 1: High level is being output by the timer output. When the timer output is enabled, rewrite to this register is ignored. |

27.3.8 TPBA_nTOL — TPBA_n Timer Output Level Register

This register controls the timer output level.

Access: This register can be read/written in 8-bit units.

Address: <TPBA_n_base> + 124_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|------------------------|
| | — | — | — | — | — | — | — | TPBA _n TOL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 27.14 TPBA_nTOL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TPBA _n TOL0 | Specifies the active level of the timer output. 0: High 1: Low. <ul style="list-style-type: none"> Setting of this bit is enabled when the timer output is enabled (TPBA_nTOE.TPBA_nTOE0 = 1). Setting of this bit is reflected when the timer output is started, and change of the output level is reflected at the next reload timing. |

CAUTION

This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see **Section 27.4.2, Compare Register Rewrite Operation**.

27.3.9 TPBAnCMP0 — TPBAn Period Setting Register

This is a 16-bit compare register for setting the PWM period.

Access: This register can be read/written in 16-bit units.

Address: <TPBAn_base> + 100_H

Value after reset: 0000_H

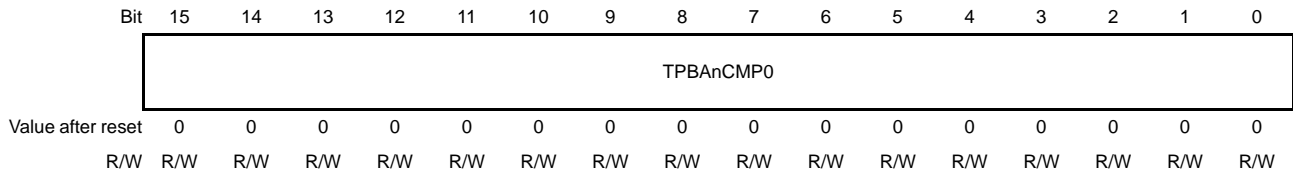


Table 27.15 TPBAnCMP0 Register Setting

| Operating Mode | PWM Period | Minimum Value (Period) | Maximum Value (Period) |
|----------------|---------------|------------------------|------------------------|
| 8 bits | TPBAnCMP0 + 1 | 1 | 100 _H |
| 16 bits | TPBAnCMP0 + 1 | 1 | 10000 _H |

CAUTION

- The PWM period is (TPBAnCMP0 + 1) count clock periods. Accordingly, for PWM output with 100% duty cycle, the maximum settable value is FFFE_H (FE_H).
- This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see **Section 27.4.2, Compare Register Rewrite Operation**.

27.3.10 TPBAnBUFm — TPBAn Duty Setting Register

This register is a 16×64 buffer register for duty setting.

Access: This register can be read/written in 16-bit units.

Address: <TPBAn_base> + m × 4_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | TPBAnBUFm | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 27.16 TPBAnBUFm Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------------|---|
| 15 to 0 | TPBAnBUFm15 to TPBAnBUFm0 | Sets the duty value. This register can set the duty value either in 16 bits × 64 patterns mode (TPBAnDPS = 0) or 8 bits × 128 patterns mode (TPBAnDPS = 1) by setting the TPBAnDPS bit. In either mode, this register is accessed in 16-bit units by the CPU. For details, see Section 27.4.3, Duty Rewrite Operation . |

CAUTION

The value set to this register is transferred to the duty setting buffer register (TPBAnCB2) synchronously with a period-matched detection interrupt (INTTPBAnIPRD). Rewrite during timer operation is reflected immediately. For details, see **Section 27.4.3, Duty Rewrite Operation**.

- When duty setting register with 8 bits × 128 patterns is used, the duty is set in the range from 00_H to FF_H.

The formula to output a waveform of duty 100% is: $TPBAnBUFm = TPBAnCMP0 + 1 \leq 00FF_H$.

Therefore, when PWM output of duty 100% is required, the maximum value of TPBAnCMP0 is 00FE_H. When TPBAnBUFm is greater than TPBAnCMP0 + 1, duty value exceeds 100%, but the output is restricted to 100%.

- When duty setting register with 16 bits × 64 patterns is used, the duty is set in the range from 0000_H to FFFF_H.

The formula to output a waveform of duty 100% is: $TPBAnBUFm = TPBAnCMP0 + 1 \leq FFFF_H$.

Therefore, when PWM output of duty 100% is required, the maximum value of TPBAnCMP0 is FFFE_H.

When TPBAnBUFm is greater than TPBAnCMP0 + 1, the duty value exceeds 100%, but the output is restricted to 100%.

27.3.11 TPBAnCMP1 — TPBAn Pattern Number Setting Register

This register sets the number of PWM output patterns.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 104_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|-----------|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | TPBAnCMP1 | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 27.17 TPBAnCMP1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|--|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | TPBAnCMP1 [6:0] | Sets the number of patterns within the following range. TPBAnDPS = 0: 0 to 63 TPBAnDPS = 1: 0 to 127 |

CAUTION

- This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see **Section 27.4.2, Compare Register Rewrite Operation**.
- If 64 or a greater number is set as the number of patterns when the duty setting pattern is in 16 bits × 64 patterns mode (TPBAnDPS = 0), the address pointer changes from 63 to 00, and the duty value is transferred from 00 again. A number-of-patterns matched detection interrupt signal (INTTPBAnIPAT) is output by the match of the specified number of patterns and the lower 7-bit values of TPBAnCNT1.

27.3.12 TPBAncNT0 — TPBAnc Timer Counter Register

This register is a timer counter register that generates PWM output.

Access: This register can only be read in 16-bit units.

Address: <TPBAnc_base> + 108_H

Value after reset: FFFF_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| TPBAncNT0 | | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

16-bit counter

This register is a counter register through which the 16-bit counter value can be read.

27.3.13 TPBAncNT1 — TPBAnc Address Counter Register

This register is a counter register that indicates the address pointer to the duty setting register.

Access: This register can only be read in 8-bit units.

Address: <TPBAnc_base> + 10C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---|
| TPBAncNT1 | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

7-bit counter

This register is a counter register that indicates the address of the TPBAncBUFm register.

27.3.14 TPBA_nTE — TPBA_n Enable Status Register

This register indicates whether the timer counter is operating or stopped.

Access: This register can only be read in 8-bit units.

Address: <TPBA_n_base> + 128_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|-----------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TPBA _n TE0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 27.18 TPBA_nTE Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | TPBA _n TE0 | Indicates whether the timer counter is operating or stopped. 0: The timer counter is stopped. 1: The timer counter is operating. <ul style="list-style-type: none"> The TPBA_nTE0 bit is set to 1 when 1 is written to the TPBA_nTS bit or when a simultaneous start trigger is input. The TPBA_nTE0 bit is cleared to 0 when 1 is written to the TPBA_nTT bit. |

27.3.15 TPBA_nTS — TPBA_n Start Trigger Register

This register controls the timer counter start trigger.

Access: This register can only be written in 8-bit units. It is always read as 0.

Address: <TPBA_n_base> + 12C_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|-----------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | TPBA _n TS0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 27.19 TPBA_nTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TPBA _n TS0 | This bit is a trigger bit that enables the timer counter. 0: Write access is ignored. 1: Starts counting (TPBA _n TE = 1). |

CAUTION

Write access to this register during counting (TPBA_nTE = 1) is ignored.

27.3.16 TPBA_nTT — TPBA_n Stop Trigger Register

This register controls the timer counter stop trigger.

Access: This register can only be written in 8-bit units. It is always read as 0.

Address: <TPBA_n_base> + 130_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-----------------------|
| | — | — | — | — | — | — | — | TPBA _n TT0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 27.20 TPBA_nTT Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | TPBA _n TT0 | This bit is a trigger bit that disables the timer counter. 0: Write access is ignored. 1: Disables counting (TPBA _n TE = 0). |

27.4 Function

27.4.1 Basic Operation

27.4.1.1 Basic Operation of 16-Bit Counter (TPBAnCNT0)

Counting start

The 16-bit counter (TPBAnCNT0) starts counting from the value after reset $FFFF_H$.

Counter clear

The 16-bit counter is cleared by the match of the counter value and the buffer register (TPBAnCB0) set value of TPBAnCMP0.

Counter read during counting

The 16-bit counter value during counting can be read through TPBAnCNT0.

27.4.1.2 Basic Operation of 7-Bit Counter (TPBAnCNT1)

Counting start

The 7-bit counter (TPBAnCNT1) is initialized to 00_H and starts counting. Subsequently, the counter value is incremented synchronously with a period-matched detection interrupt (INTTPBAnIPRD).

Counter clear

The 7-bit counter is cleared by the match of the counter value and the buffer register (TPBAnCB1) set value of TPBAnCPM1.

Counter read during counting

The 7-bit counter value during counting can be read through TPBAnCNT1. The read value indicates TPBAnBUFm in which the duty value to be transferred next is stored.

27.4.2 Compare Register Rewrite Operation

The following registers are rewritten by reload.

- TPBAnCMP0
- TPBAnCMP1
- TPBAnTOL

Reload mode (simultaneous rewrite function)

Writing to TPBAnRDT enables reload of the registers corresponding to the set bits (sets the reload request flag (TPBAnRSF.TPBAnRSFk)), and the values of all the registers to be reloaded are updated simultaneously at the next reload timing (reload).

The reload timing of TPBAnCMP0 and TPBAnTOL is set by TPBAnRDM.

The reload timing of TPBAnCMP1 is the match timing (INTTPBAnIPAT) of the 7-bit counter (TPBAnCNT1) and the buffer register (TPBAnCB1) of TPBAnCMP1.

The registers to be reloaded should be rewritten when the reload request flag (TPBAnRFS.TPBAnRSFk) is 0.

Note: k = 0, 1

Setting Flow for Registers to Be Reloaded

The rewritten values of the registers to be reloaded (TPBAnCMP0, TPBAnCMP1, and TPBAnTOL) can be transferred to the respective buffer registers simultaneously at the reload timing.

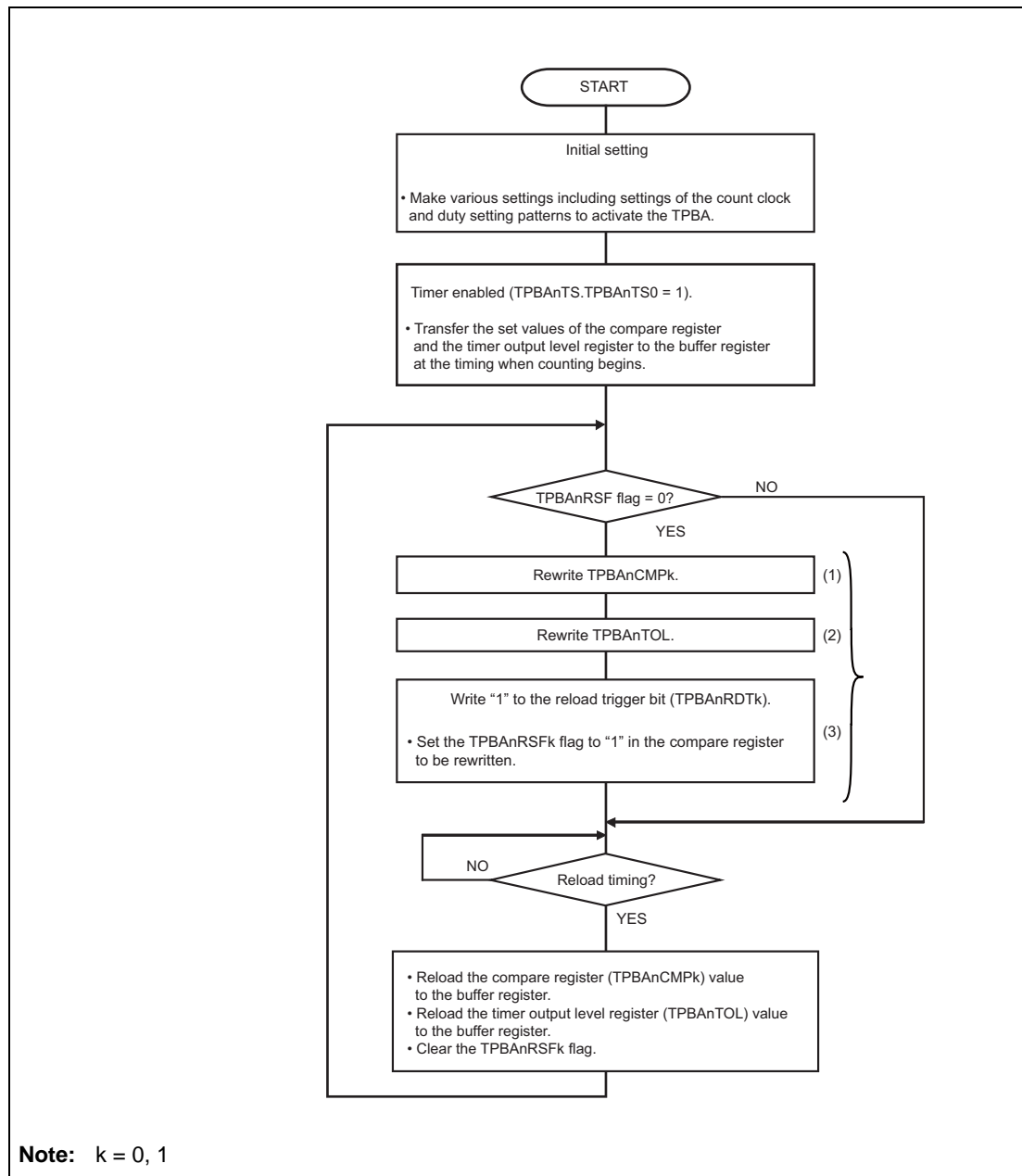


Figure 27.2 Basic Operation Flow of Reload (Simultaneous Rewrite Function)

CAUTION

Setting the TPBAnRDTk bit to 1 enables reload. Accordingly, the TPBAnRDTk bit should be rewritten after the registers to be reloaded have been rewritten.

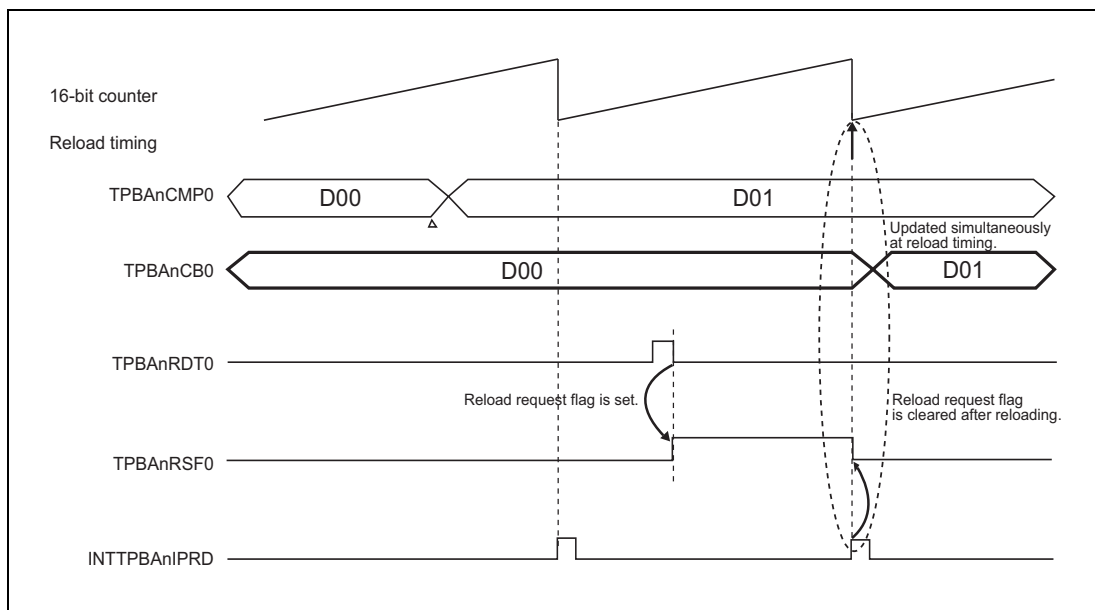


Figure 27.3 Simultaneous Rewrite Timing (TPBAnDPS = 0, TPBAnRDM = 0, and TPBAnTOL = 0)

27.4.3 Duty Rewrite Operation

TPBAnBUFm can be rewritten during operation.

The rewritten setting is reflected immediately.

27.4.3.1 TPBAnBUFm Setting Flow

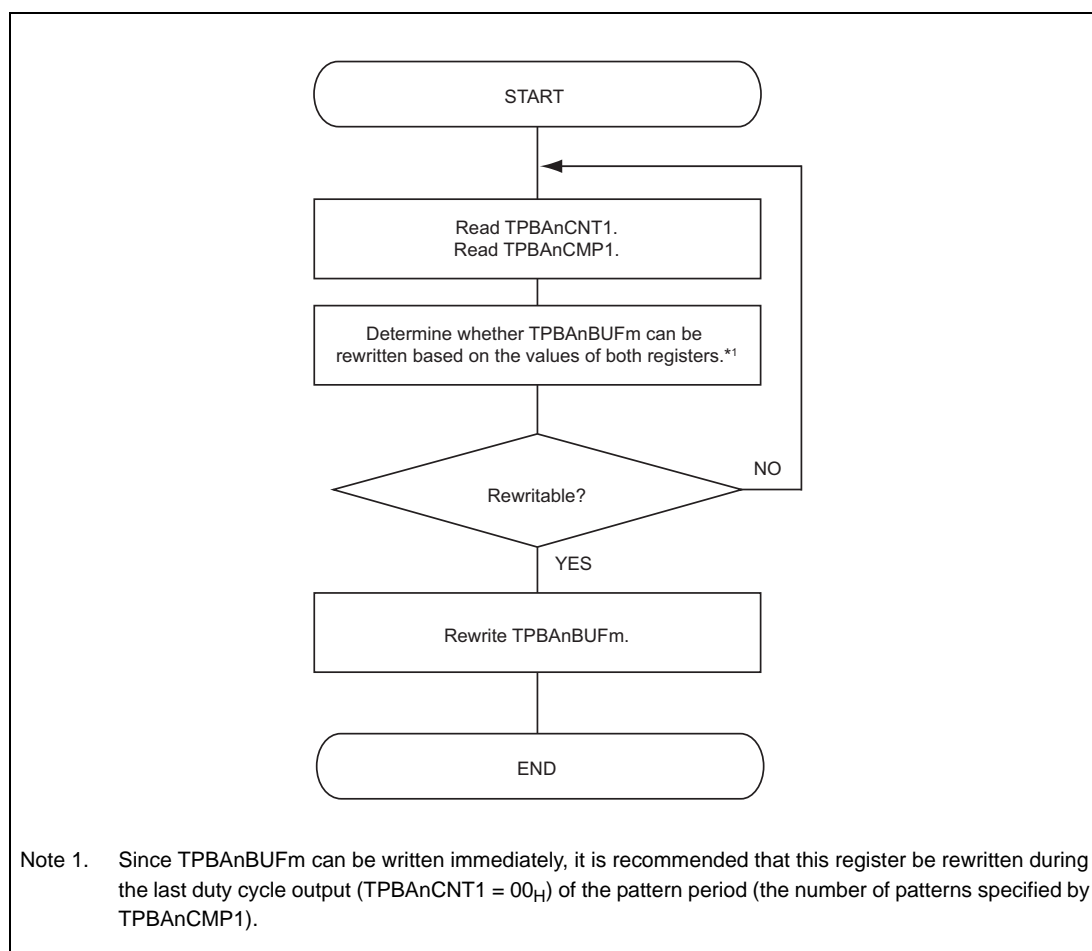


Figure 27.4 Basic Rewrite Flow of TPBAnBUFm

27.4.3.2 Access to TPBA_nBUF_m

TPBA_nBUF_m is accessed in 16 bit units. The following shows the access in 16 bits × 64 patterns mode and the access in 8 bits × 128 patterns mode.

- In 16 bits × 64 patterns mode (TPBA_nDPS = 0)
This register is accessed by the CPU in units of one 16-bit pattern.

| | | |
|------------|---|-------------------|
| 15 | 0 | |
| Pattern 64 | | 00FC _H |
| Pattern 63 | | 00F8 _H |
| : | | : |
| Pattern 3 | | 0008 _H |
| Pattern 2 | | 0004 _H |
| Pattern 1 | | 0000 _H |

- In 8 bits × 128 patterns mode (TPBA_nDPS = 1)
This register is accessed by the CPU in units of two 8-bit patterns.

| | | | | |
|-------------|---|-------------|---|-------------------|
| 15 | 8 | 7 | 0 | |
| Pattern 128 | | Pattern 127 | | 00FC _H |
| Pattern 126 | | Pattern 125 | | 00F8 _H |
| : | | : | | : |
| Pattern 6 | | Pattern 5 | | 0008 _H |
| Pattern 4 | | Pattern 3 | | 0004 _H |
| Pattern 2 | | Pattern 1 | | 0000 _H |

27.4.3.3 Relationship between TPBAncNT1 Read Value and TPBAncBUFm

The duty value of the currently output PWM waveform can be obtained by reading the TPBAncNT1 count value during operation. TPBAncBUFm in which the currently output duty value is stored can be found by one of the following formulas.

Table 27.21 TPBAncBUFm Formula

| TPBAncDPS Bit | Formula | | |
|--------------------------------------|--|--|-----------------------------|
| | TPBAncNT1 \neq 00 _H | | TPBAncNT1 = 00 _H |
| 0: 16 bits \times 64 patterns mode | TPBAncNT1 - 01 _H ⁽¹⁾ | | TPBAncCMP1 ⁽²⁾ |
| 1: 8 bits \times 128 patterns mode | TPBAncNT1 value is an odd number | TPBAncNT1/2 ⁽³⁾ | TPBAncCMP1/2 ⁽⁵⁾ |
| | TPBAncNT1 value is an even number | (TPBAncNT1 / 2) - 01 _H ⁽⁴⁾ | |

- (1) When TPBAncDPS = 0 and the TPBAncNT1 \neq 00_H
The applicable register is found by the formula TPBAncNT1 - 01_H.
(Example) When TPBAncNT1 = 08_H: 08_H - 01_H = 07_H \rightarrow TPBAncBUF07
- (2) When TPBAncDPS = 0 and the TPBAncNT1 = 00_H
The applicable register is found by the TPBAncCMP1 value.
(Example) When TPBAncCMP1 = 08_H: TPBAncBUF08
- (3) When TPBAncDPS = 1 and the TPBAncNT1 = an odd number
The applicable register is found by the formula TPBAncNT1 / 2
(Example) When TPBAncNT1 = 07_H: 07_H / 02_H = 03_H \rightarrow TPBAncBUF03 (lower 8 bits)
- (4) When TPBAncDPS = 1 and the TPBAncNT1 = an even number
The applicable register is found by the formula (TPBAncNT1 / 2) - 01_H.
(Example) When TPBAncNT1 = 08_H: (08_H / 02_H) - 01_H = 03_H \rightarrow TPBAncBUF03 (upper 8 bits)
- (5) When TPBAncDPS = 1 and the TPBAncNT1 = 00_H
The applicable register is found by the formula TPBAncCMP1 / 2.
(Example) When TPBAncCMP1 = 08_H: 08_H / 2 = 04_H \rightarrow TPBAncBUF04 (lower 8 bits)

27.4.4 Basic Operation Example

Overview

A PWM signal is output from the TPBA_nO pin according to the PWM period set in the TPBA_nCMP0 register and duty cycle set in the TPBA_nBUF00 to TPBA_nBUF63 registers.

Prerequisites

- Select 16 bits × 64 patterns mode or 8 bits × 128 patterns mode by setting TPBA_nDPS.
- Set the duty cycle to TPBA_nBUF00 to TPBA_nBUF63.
- Set the number of patterns to TPBA_nCMP1.

Functional description

Set the PWM period, the number of patterns, duty cycle, and level to be output. Set TPBA_nTS.TPBS_nTS0 = 1 (or input a simultaneous start trigger) to start incrementing the timer counter value.

The TPBA_nO output is set to the active level at the same time the counting begins. TPBA_nCNT1 is incremented, and points to the address of the buffer in which the subsequent duty value is stored.

The output is set to the inactive level by the match of the 16-bit counter and the TPBA_nBUF_m buffer register (TPBA_nCB2).

The duty value is then transferred from TPBA_nBUF_m to the buffer register (TPBA_nCB2) by the match of the 16-bit counter and the TPBA_nCMP0 buffer register (TPBA_nCB0). Then, TPBA_nCNT1 is incremented, and a period-matched detection interrupt (INTTPBA_nIPRD) is generated. The TPBA_nO output is set to the active level after one count clock.

During counting, a duty-cycle-matched detection interrupt (INTTPBA_nIDTY) is generated by the match of the 16-bit counter and the buffer register (TPBA_nCB2) of TPBA_nBUF_m.

A number-of-patterns matched detection interrupt (INTTPBA_nIPAT) is generated by the match of the 7-bit counter and the TPBA_nCMP1 buffer register (TPBA_nCB1).

27.4.4.1 List of Operations

Table 27.22 16-Bit Counter Function

| Operation | | Setting Condition |
|----------------|-------|--|
| 16-bit counter | Start | Writing 1 to TPBA _n TS or set simultaneous start trigger. |
| | Clear | Compare match of TPBA _n CMP0 buffer register and 16-bit counter |
| | Stop | Writing 1 to TPBA _n TT |

Table 27.23 7-Bit Counter Function

| Operation | | Setting Condition |
|---------------|-------|---|
| 7-bit counter | Start | Writing 1 to TPBA _n TS or set simultaneous start trigger. |
| | Clear | Compare match of TPBA _n CMP1 buffer register and 7-bit counter |
| | Stop | Writing 1 to TPBA _n TT |

Table 27.24 Functions of Compare Registers and Buffer Registers

| Register (Data) | Buffer Register | Rewrite Method | Rewrite during Operation | Function |
|------------------------------------|------------------------|---------------------|--------------------------|----------------------------|
| TPBA _n CMP0 | TPBA _n CB0 | Reload | Possible | Setting period |
| TPBA _n CMP1 | TPBA _n CB1 | Reload | Possible | Setting number of patterns |
| TPBA _n BUF _m | TPBA _n CB2 | Rewrite immediately | Possible | Setting duty |
| TPBA _n TOL | TPBA _n TOLB | Reload | Possible | Setting output level |

Buffer Registers

The registers that specify period, the number of patterns, duty, and timer output level consist of data registers that a user can directly set and buffer registers that a user cannot directly set.

Table 27.25 Timer Output Function

| Pin | Function |
|---------------------|--|
| TPBA _n O | <ul style="list-style-type: none"> When output is enabled (TPBA_nTOE = 01_H) PWM output by compare match of the TPBA_nBUF_m buffer register (TPBA_nCB2) and the 16-bit counter When output is disabled (TPBA_nTOE = 00_H) TPBA_nTO set value |

Table 27.26 Interrupt Requests

| Interrupt | Function |
|---------------------------|--|
| INTTPBA _n IPRD | Period-matched detection interrupt |
| INTTPBA _n IDTY | Duty-cycle-matched detection interrupt |
| INTTPBA _n IPAT | Number-of-patterns matched detection interrupt |

Table 27.27 Compare Match Timing

| Compare Match | Timing |
|------------------------------------|---|
| TPBA _n CMP0 | When the 16-bit counter changes from match with TPBA _n CMP0 to 0000 _H . |
| TPBA _n CMP1 | When the 7-bit counter changes from match with TPBA _n CMP1 to 01 _H . |
| TPBA _n BUF _m | When the 16-bit counter matches with the buffer register (TPBA _n CB2). |

Table 27.28 Example of Setting Each Timer Output Condition

| Pin | Item | Output Period | Output Duty | |
|--------|------------|--|--|---|
| | | | Output Condition | Setting Condition |
| TPBAnO | PWM output | $(\text{TPBAnCMP0} + 1) \times$ count clock | Outputs an inactive level throughout one period (duty cycle 0%). | $\text{TPBAnBUFm} = 0000_{\text{H}}$ |
| | | | Outputs an active level of one count clock in one period. | $\text{TPBAnBUFm} = 0001_{\text{H}}$ |
| | | | Outputs an inactive level of one count clock in one period | $\text{TPBAnBUFm} =$ TPBAnCMP0 |
| | | | Outputs an active level throughout one period (duty cycle 100%). | $\text{TPBAnBUFm} \geq$ $\text{TPBAnCMP0} + 1$ |

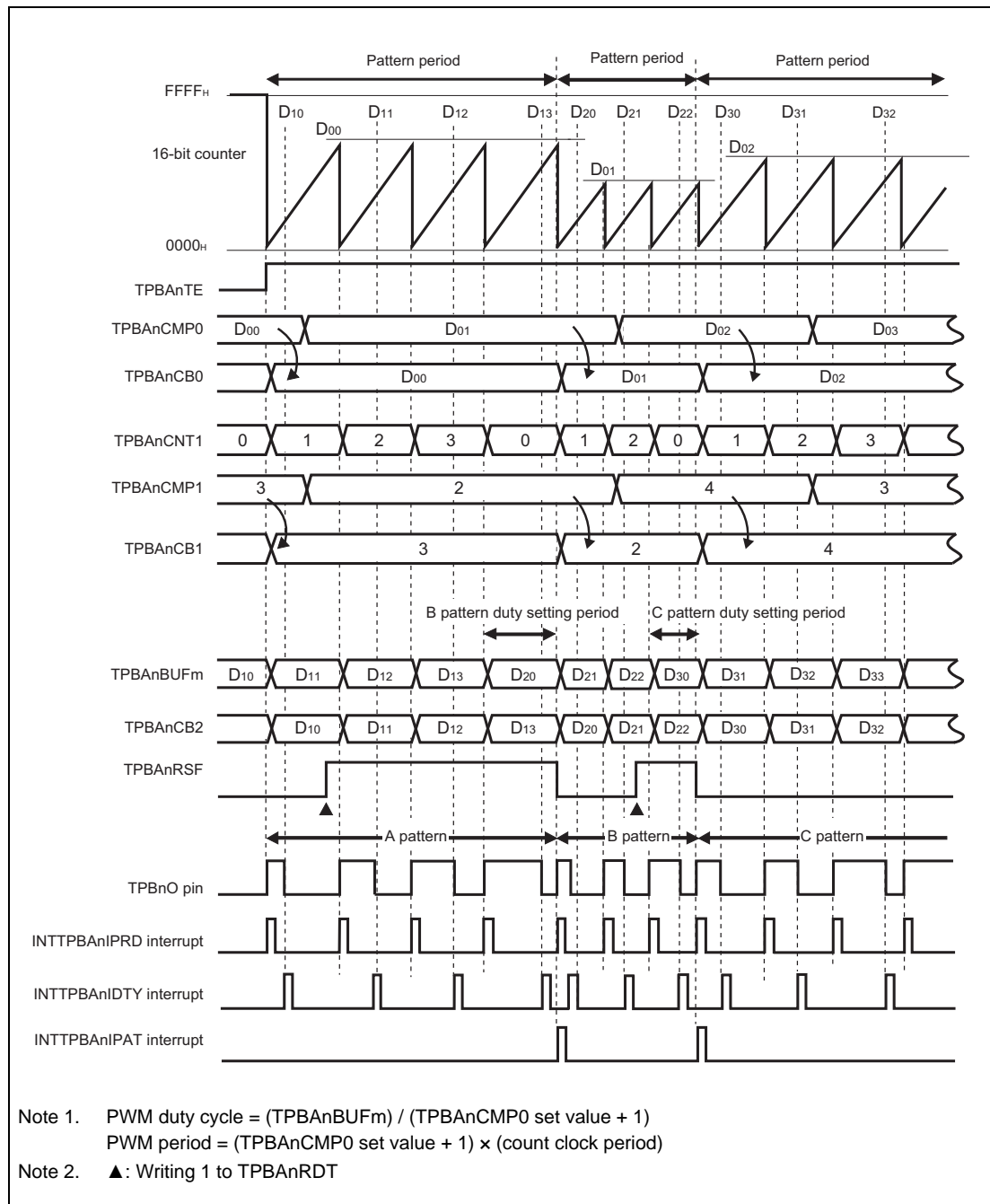


Figure 27.5 Example of Basic Timing (1/2)

CAUTION

TPBAnO outputs active level 1 count clock after output of INTTPBAnIPRD and outputs inactive level at INTTPBAnIDTY output timing.

When a number-of-patterns matched detection interrupt is used as a trigger of the TPBAnCMP0 and TPBAnTOL reload timing (TPBAnIRDM.TPBAnRDM0 = 0 and TPBAnTOL = 0)

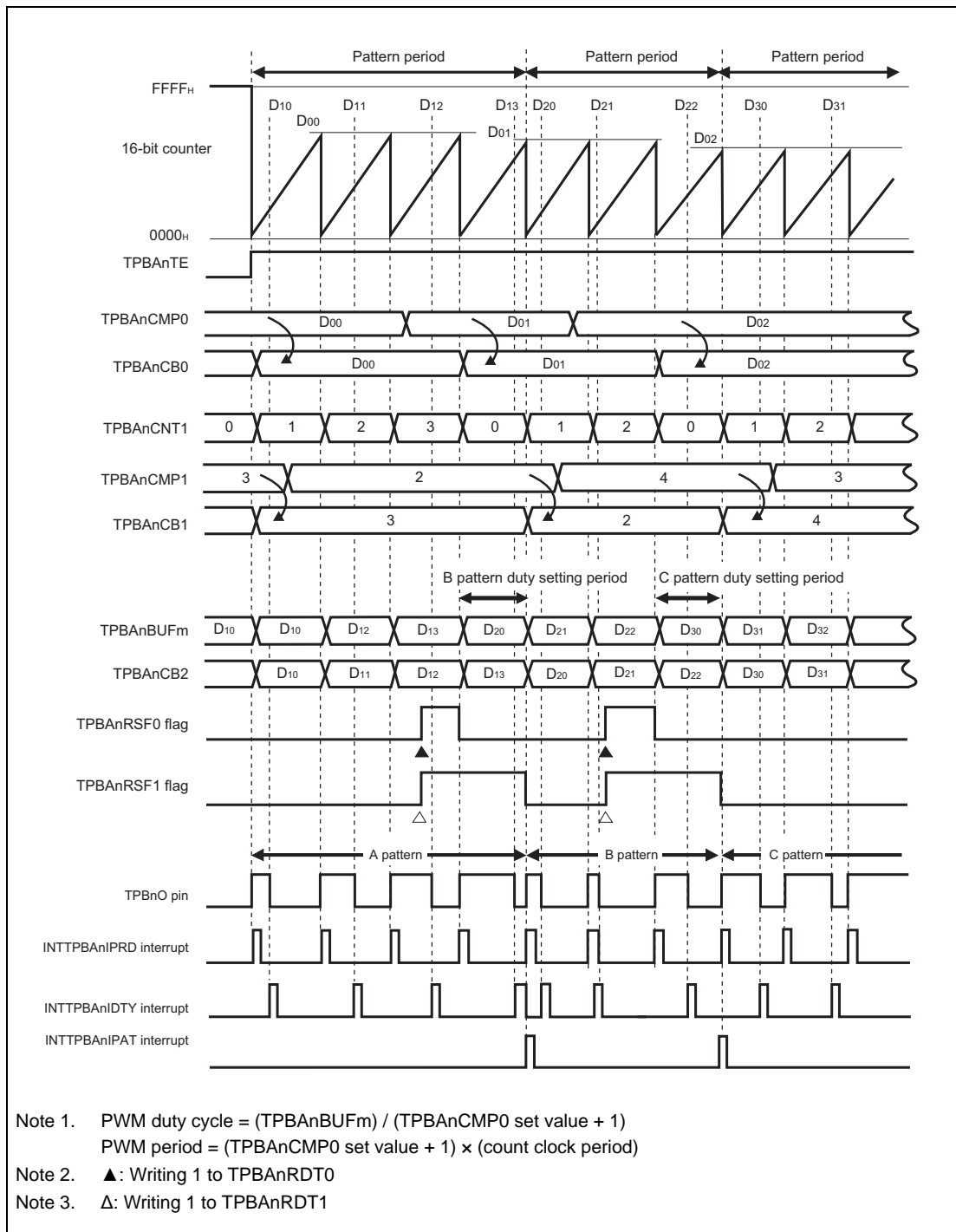


Figure 27.6 Example of Basic Timing (2/2)

CAUTION

TPBA_nO outputs active level 1 count clock after output of INTTPBAAnIPRD and outputs inactive level at INTTPBAAnIDTY output timing.

When a period-matched detection interrupt is used as a trigger of the TPBAAnCMP0 and TPBAAnTOL reload timing (TPBAAnIRDM.TPBAAnRDM0 = 1 and TPBAAnTOL = 0)

Section 28 Encoder Timer (ENCA)

The Encoder Timer (ENCA) operates the timer counter with counter up/down control and clear control by encoder inputs. ENCA is implemented 2 units.

28.1 Features of RH850/P1M-E ENCA

28.1.1 Units and Channels

This microcontroller has the following number of ENCA units.

Each ENCA unit has one channel ENCA. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 28.1 Number of Units

| Product | RH850/P1M-E 100 pins (eVR) | RH850/P1M-E 100 pins (DPS) | RH850/P1M-E 144 pins (eVR) | RH850/P1M-E 144 pins (DPS) |
|-----------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| Number of Units | 2 | | | |
| Name | ENCA _n (n = 0, 1) | | | |

Table 28.2 Unit Configurations and Channels

| Unit Name (Channel Name) ENCA _n | Channels per Unit | RH850/P1M-E 100 pins (eVR) | RH850/P1M-E 100 pins (DPS) | RH850/P1M-E 144 pins (eVR) | RH850/P1M-E 144 pins (DPS) |
|--|----------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| ENCA0 | 1 | √ | √ | √ | √ |
| ENCA1 | 1 | √ | √ | √ | √ |

Table 28.3 Index

| Index | Description |
|-------|--|
| n | Throughout this section, the individual ENCA units are identified by the index “n” (n = 0, 1); for example, ENCA _n CTL is the ENCA _n control register. |

28.1.2 Register Base Address

ENCA_n base addresses are listed in the following table.

ENCA_n register addresses are given as offsets from the base addresses.

Table 28.4 Register Base Addresses

| Base Address Name | Base Address |
|-------------------|------------------------|
| <ENCA0_base> | FFE8 0000 _H |
| <ENCA1_base> | FFE8 1000 _H |

28.1.3 Clock Supply

Clock supply by and to ENCA_n is listed in the following table.

Table 28.5 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-------------------|-----------------|-------------------------------------|
| ENCA _n | PCLK | High-speed peripheral clock CLK_HSB |

28.1.4 Interrupt Requests

ENCA_n interrupt requests are listed in the following table.

Table 28.6 Interrupt Requests

| Unit Interrupt Name | Outline | Interrupt Number | DMA/DTS Trigger Number | Others |
|---------------------|---|------------------|------------------------|--------|
| ENCA0 | | | | |
| INTENCA0IOV | ENCA0 overflow interrupt | 276 | — | — |
| INTENCA0I0 | ENCA0 capture/compare match interrupt 0 | 277 | 107 | — |
| INTENCA0I1 | ENCA0 capture/compare match interrupt 1 | 278 | 108 | PIC2 |
| INTENCA0IUD | ENCA0 underflow interrupt | 279 | — | PIC1 |
| INTENCA0IEC | ENCA0 encoder clear interrupt | 280 | — | PIC1 |
| ENCA1 | | | | |
| INTENCA1IOV | ENCA1 overflow interrupt | 281 | — | — |
| INTENCA1I0 | ENCA1 capture/compare match interrupt 0 | 282 | 109 | — |
| INTENCA1I1 | ENCA1 capture/compare match interrupt 1 | 283 | 110 | PIC2 |
| INTENCA1IUD | ENCA1 underflow interrupt | 284 | — | — |
| INTENCA1IEC | ENCA1 encoder clear interrupt | 285 | — | PIC1 |

28.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

28.1.6 External Input/Output Signals

External input/output signals of ENCA_n are listed below.

Table 28.7 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Pin Signal Name |
|------------------|-----|---|----------------------------------|
| ENCA0 | | | |
| ENCA0TTIN0 | I | ENCA0 capture trigger input 0 ^{*1} | ENCA0TIN0 ^{*2} |
| ENCA0TTIN1 | I | ENCA0 capture trigger input 1 ^{*1} | ENCA0TIN1 ^{*2} |
| ENCA0E0 | I | ENCA0 encoder input 0 ^{*1} | ENCA0E0 ^{*2} |
| ENCA0E1 | I | ENCA0 encoder input 1 ^{*1} | ENCA0E1 ^{*2} |
| ENCA0EC | I | ENCA0 encoder clear input ^{*1} | ENCA0EC ^{*2} |
| ENCA1 | | | |
| ENCA1TTIN0 | I | ENCA1 capture trigger input 0 ^{*1} | ENCA1TIN0 ^{*1} |
| ENCA1TTIN1 | I | ENCA1 capture trigger input 1 ^{*1} | ENCA1TIN1 ^{*2} |
| ENCA1E0 | I | ENCA1 encoder input 0 ^{*1} | ENCA1E0 ^{*2} |
| ENCA1E1 | I | ENCA1 encoder input 1 ^{*1} | ENCA1E1 ^{*2} |
| ENCA1EC | I | ENCA1 encoder clear input ^{*1} | ENCA1EC ^{*2} |

Note 1. Setting of the noise filter for the port is required when the input pin is used. For details, see **Section 2.6, Noise Filter and Edge Level Detection Circuit**.

Note 2. The input signal can be selected by the PIC function. For details, see **Section 29.2.2.23, PIC1AREG30 — Timer Input/Output Control Register 30**.

28.1.7 Internal Input/Output Signals

Table 29.8 shows the input/output signals connecting ENCA and PIC.

Table 28.8 Internal Input/Output Signals

| Unit Signal Name | Description | Connected to |
|------------------|---|--------------|
| ENCATSST | Simultaneous start trigger | PIC |
| ENCATTIN1 | ENCA _n capture trigger input 1 | PIC |

28.2 Overview

28.2.1 Functional Overview

- Generation of the counter control signal from the encoder input signal, and performs count operation in synchronization with PCLK.
- Capture function for capturing the counter value with an external trigger signal
- Compare function for compare match judgment with the counter value
- Two capture/compare registers that can be set separately for capture operation and for compare operation
- Interrupt mask function for masking the interrupt request signal output as a result of compare match judgment during compare operation
- Function for loading the value of the capture/compare register to the counter upon underflow occurrence
- Encoder input signal can be used as the timer counter clear condition
- Edge or level can be selected for determining the presence of the encoder input signal that is used as the timer counter clear condition
- Detection of counter overflow and underflow and output of error flags and error interrupts
- Five interrupts: 2 capture/compare interrupts, 1 counter clear interrupt, 1 overflow interrupt, and 1 underflow interrupt

28.2.2 Block Diagram

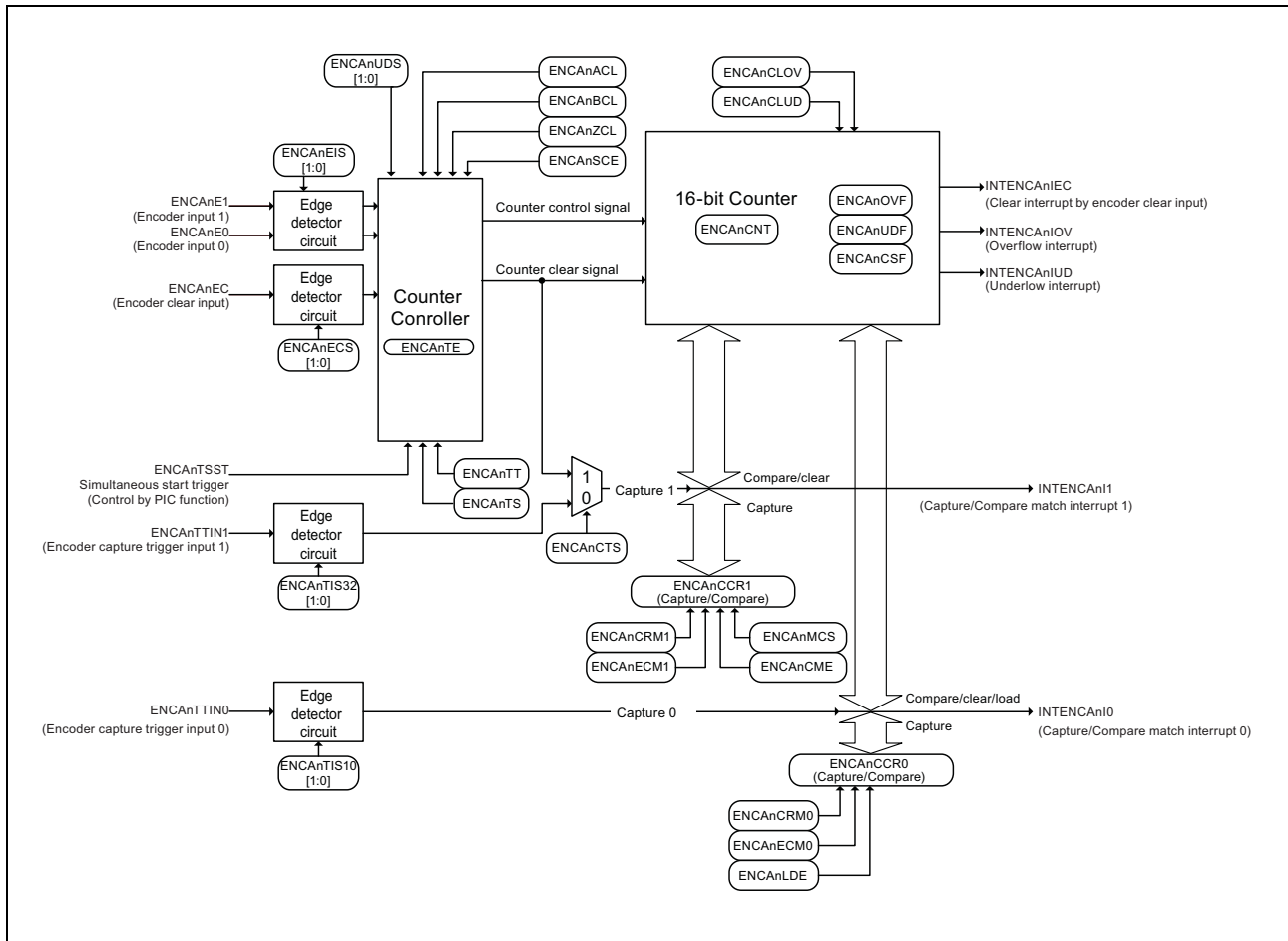


Figure 28.1 Encoder Timer Block Diagram

28.3 Registers

28.3.1 List of Registers

ENCA registers are listed in the following table.

For details about <ENCA_base>, see **Section 28.1.2, Register Base Address**.

Table 28.9 Registers

| Module | Register | Symbol | Address |
|-------------------|-----------------------------------|------------------------|---|
| ENCA _n | ENCA capture/compare register 0 | ENCA _n CCR0 | <ENCA _n _base> |
| ENCA _n | ENCA capture/compare register 1 | ENCA _n CCR1 | <ENCA _n _base> + 04 _H |
| ENCA _n | ENCA counter register | ENCA _n CNT | <ENCA _n _base> + 08 _H |
| ENCA _n | ENCA status flag register | ENCA _n FLG | <ENCA _n _base> + 0C _H |
| ENCA _n | ENCA status flag clear register | ENCA _n FGC | <ENCA _n _base> + 10 _H |
| ENCA _n | ENCA timer enable status register | ENCA _n TE | <ENCA _n _base> + 14 _H |
| ENCA _n | ENCA timer start trigger register | ENCA _n TS | <ENCA _n _base> + 18 _H |
| ENCA _n | ENCA timer stop trigger register | ENCA _n TT | <ENCA _n _base> + 1C _H |
| ENCA _n | ENCA I/O control register 0 | ENCA _n IOC0 | <ENCA _n _base> + 20 _H |
| ENCA _n | ENCA control register | ENCA _n CTL | <ENCA _n _base> + 40 _H |
| ENCA _n | ENCA I/O control register 1 | ENCA _n IOC1 | <ENCA _n _base> + 44 _H |

28.3.2 ENCACTL — ENCA Control Register

This register is used to configure various operation settings of the Encoder Timer.

Access: This register can be read/written in 16-bit units.
Writing to this register during operation is prohibited.

Address: <ENCA_n_base> + 40_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------------------|-----------------------|----|----|----|----|------------------------|------------------------|-----------------------|---|---|-----------------------|------------------------|------------------------|-----------------------------|-----|
| | ENCA _n CME | ENCA _n MCS | — | — | — | — | ENCA _n CRM1 | ENCA _n CRM0 | ENCA _n CTS | — | — | ENCA _n LDE | ENCA _n ECM1 | ENCA _n ECM0 | ENCA _n UDS [1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R | R | R/W | R/W | R/W | R/W | R/W |

Table 28.10 ENCACTL Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 15 | ENCA _n CME | Encoder Clear Mask Enable This bit is used to enable/disable masking of compare match interrupt detection when the compare function is used. 0: Disables the compare match interrupt (INTENCA _n I1) mask function for the ENCA _n CCR1 register 1: Enables the compare match interrupt (INTENCA _n I1) mask function for the ENCA _n CCR1 register. This bit is valid only when ENCA _n CRM1 = 0. When this bit is set to 1, setting ENCA _n ECM1 to 1 is prohibited. |
| 14 | ENCA _n MCS | Encoder Mask Clear Select This bit is used to select the trigger for cancelling masking of compare match interrupt detection INTENCA _n I1 when the compare function is used. This bit is valid only when ENCA _n CRM1 = 0. 0: Masking of compare match interrupt detection is cancelled when the ENCA _n CCR1 register is written. 1: Masking of compare match interrupt detection is cancelled when one of the following three operations is performed. – Encoder clear input operation – Timer counter clear operation upon compare match between ENCA _n CNT and ENCA _n CCR0 when ENCA _n ECM0 = 1 – Loading from ENCA _n CCR0 to timer counter upon underflow detection when ENCA _n LDE = 1 |
| 13 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9 | ENCA _n CRM1 | ENCA _n CCR1 Register Mode 0: ENCA _n CCR1 used as compare register. 1: ENCA _n CCR1 used as capture register. |
| 8 | ENCA _n CRM0 | ENCA _n CCR0 register mode bit 0: ENCA _n CCR0 used as compare register. 1: ENCA _n CCR0 used as capture register. |
| 7 | ENCA _n CTS | ENCA _n CCR1 Capture Trigger Select This is a trigger selection bit for the capture operation to the ENCA _n CCR1 register. This bit is valid only when ENCA _n CRM1 = 1. 0: Uses ENCA _n TTIN1 of capture trigger 1 signal as the trigger for capturing to the ENCA _n CCR1 register. 1: Uses the encoder clear signal selected by ENCA _n SCE as the capture trigger to the ENCA _n CCR1 register. |
| 6, 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 28.10 ENCA_nCTL Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------------------------|---|
| 4 | ENCA _n LDE | <p>ENCA_n Counter Load Enable</p> <p>This bit is used to enable/disable setting value loading to the counter upon underflow occurrence.</p> <p>This bit is valid only when ENCA_nCRM0 = 0.</p> <p>When ENCA_nCRM0 = 1, loading of the ENCA_nCCR0 register setting value to the counter upon occurrence of an underflow is not performed, regardless of the value of this bit.</p> <p>0: Disable loading of ENCA_nCCR0 register setting value to counter upon occurrence of a counter underflow.</p> <p>1: Enable loading of ENCA_nCCR0 register setting value to counter upon occurrence of a counter underflow.</p> |
| 3 | ENCA _n ECM1 | <p>Encoder Clear Mode 1</p> <p>This bit is used to set the counter clear operation upon match between the counter value and ENCA_nCCR1 setting value.</p> <p>This bit is valid only when ENCA_nCRM1 = 0.</p> <p>0: Does not clear the counter to 0000_H upon match of timer counter value and ENCA_nCCR1 setting value.</p> <p>1: Clears the counter to 0000_H upon match of timer counter value and ENCA_nCCR1 setting value if the next count is a down-count.</p> |
| 2 | ENCA _n ECM0 | <p>Encoder Clear Mode 0</p> <p>This bit is used to set the counter clear operation upon match between the counter value and ENCA_nCCR0 setting value.</p> <p>This bit is valid only when ENCA_nCRM0 = 0.</p> <p>0: Does not clear the counter to 0000_H upon match of timer counter value and ENCA_nCCR0 setting value.</p> <p>1: Clears the counter to 0000_H upon match of timer counter value and ENCA_nCCR0 setting value if the next count is a up-count.</p> |
| 1, 0 | ENCA _n UDS[1:0] | <p>Up/down Count Selection 1 and 0</p> <p>These are the counter up/down control bits that use ENCA_nE0 and ENCA_nE1.</p> <p>00: Upon detection of valid edge of ENCA_nE0, - down-count when ENCA_nE1 = H, - up-count when ENCA_nE1 = L</p> <p>01: Upon detection of valid edge of ENCA_nE0, up-count, Upon detection of valid edge of ENCA_nE1, down-count</p> <p>10: At rising edge of ENCA_nE0, down-count At falling edge of ENCA_nE0, up-count However, count operation is performed only when ENCA_nE1 = L.</p> <p>11: Detection of both edges of ENCA_nE0 and ENCA_nE1. Judgment of count operation combining both detected edge and level.</p> |

28.3.3 ENCA_nIOC0 — ENCA I/O Control Register 0

This register is used to select the input edge of capture triggers 0 and 1 (ENCA_nTTIN0, ENCA_nTTIN1).

Access: This register can be read/written in 8-bit units.

Address: <ENCA_n_base> + 20_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------------------------|-----|------------------------------|-----|
| | — | — | — | — | ENCA _n TIS32[1:0] | | ENCA _n TIS10[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 28.11 ENCA_nIOC0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------------------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3, 2 | ENCA _n TIS32 [1:0] | Input Edge Selection for Capture Trigger 1 These bits are valid only when the ENCA _n CTL register's ENCA _n CRM1 = 1 and ENCA _n CTS = 0. All other settings of ENCA _n CRM1 and ENCA _n CTS are invalid. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection |
| 1, 0 | ENCA _n TIS10 [1:0] | Input Edge Selection for Capture Trigger 0 These bits are valid only when ENCA _n CTL.ENCA _n CRM0 = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection |

28.3.4 ENCAIOC1 — ENCA I/O Control Register 1

This register is used to perform the clear condition setting and edge selection upon encoder input.

Access: This register can be read/written in 8-bit units.
Writing to this register during operation is prohibited.

Address: <ENCA_n_base> + 44_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------|-----|----------------------------|-----|
| | ENCA _n SCE | ENCA _n ZCL | ENCA _n BCL | ENCA _n ACL | ENCA _n ECS[1:0] | | ENCA _n EIS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 28.12 ENCAIOC1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------------|---|
| 7 | ENCA _n SCE | Encoder Special-clear Enable This is an encoder special clear enable bit. When setting this bit to 1, set ENCA _n UDS[1:0] to 10 _B or 11 _B . The operation is not guaranteed if this bit is set to 1 with ENCA _n UDS[1:0] set to 00 _B or 01 _B . 0: Clears the counter upon detection of ENCA _n EC valid edge (set with ENCA _n ECS[1:0]). 1: Clears the counter upon detection of input level condition of ENCA _n EC, ENCA _n E1, and ENCA _n E0 (set with ENCA _n ZCL bit, ENCA _n BCL bit, and ENCA _n ACL bit). |
| 6 | ENCA _n ZCL | Input-Z Clear Condition Selection This bit is used to set the condition for clearing the encoder clear input (ENCA _n EC) when using the encoder special clear function. This bit is valid only when ENCA _n SCE = 1; it is invalid when ENCA _n SCE = 0. 0: Clear condition: Low level 1: Clear condition: High level |
| 5 | ENCA _n BCL | Input-B Clear Condition Selection This bit is used to set the condition for clearing the encoder input 1 (ENCA _n E1) when using the encoder special clear function. This bit is valid only when ENCA _n SCE = 1; it is invalid when ENCA _n SCE = 0. 0: Clear condition: Low level 1: Clear condition: High level |
| 4 | ENCA _n ACL | Input-A Clear Condition Selection This bit is used to set the condition for clearing the encoder input 0 input (ENCA _n E0) when using the encoder special clear function. This bit is valid only when ENCA _n SCE = 1; it is invalid when ENCA _n SCE = 0. 0: Clear condition: Low level 1: Clear condition: High level |
| 3, 2 | ENCA _n ECS[1:0] | Encoder Clear Input Edge Selection 1 and 0 These are the encoder clear input edge selection bits. These bits are valid only when ENCA _n SCE = 0; they are invalid when ENCA _n SCE = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection |
| 1, 0 | ENCA _n EIS[1:0] | Encoder Edge Input Selection 1 and 0 These are the encoder input edge selection bits. These bits are valid when ENCA _n UDS[1:0] is 00 _B or 01 _B , and are invalid when ENCA _n UDS[1:0] is 10 _B or 11 _B . 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection |

28.3.5 ENCA_nFLG — ENCA Status Flag Register

This register holds the status flags of the timer counter of ENCA_n.

Access: This register can only be read in 8-bit units.

Address: <ENCA_n_base> + 0C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|-----------------------|-----------------------|-----------------------|
| | — | — | — | — | — | ENCA _n CSF | ENCA _n UDF | ENCA _n OVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 28.13 ENCA_nFLG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 7 to 3 | Reserved | When read, the value after reset is read. |
| 2 | ENCA _n CSF | Counter Status Flag This bit reflects the current timer counter operation. This bit is cleared at the start of count operation. 0: Timer counter in up-count status 1: Timer counter in down-count status |
| 1 | ENCA _n UDF | Underflow Flag This bit reflects the occurrence of an underflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: <ul style="list-style-type: none"> – 1 is written to ENCA_nFGC.ENCA_nCLUD – ENCA_nTS is set while ENCA_nTE = 0 – The input signal of ENCA_nTSST is set to “High level” 1: This flag is set to 1 upon occurrence of an underflow during the encoder timer count operation. |
| 0 | ENCA _n OVF | Overflow Flag This bit reflects the occurrence of an overflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: <ul style="list-style-type: none"> – 1 is written to ENCA_nFGC.ENCA_nCLOV – ENCA_nTS is set while ENCA_nTE = 0 – The input signal of ENCA_nTSST is set to “High level” 1: This flag is set to 1 upon occurrence of an overflow during the encoder timer count operation. |

28.3.6 ENCAFGC — ENCA Status Flag Clear Register

This register is used to clear the timer counter status flags of ENCAFLG.

Access: This register can only be written in 8-bit units.
This register always returns 0 when read.

Address: <ENCA_n_base> + 10_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|------------------------|------------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | ENCA _n CLUD | ENCA _n CLOV |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | W | W |

Table 28.14 ENCAFGC Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|--|
| 7 to 2 | Reserved | When writing, write the value after reset. |
| 1 | ENCA _n CLUD | Underflow Flag Clear This bit clears the underflow flag. 0: Writing is ignored. 1: Clears ENCA _n UDF of the ENCA _n FLG register (clears underflow detection). |
| 0 | ENCA _n CLOV | Overflow Flag Clear This bit clears the overflow flag. 0: Writing is ignored. 1: Clears ENCA _n OVF of the ENCA _n FLG register (clears overflow detection). |

28.3.7 ENCA_nCCR0 — ENCA Capture/Compare Register 0

This register is a 16-bit capture/compare register 0.

Access: This register can only be read/written in 16-bit units.
When used as a capture register, this register is only readable. Writing to this register is ignored.
When used as a compare register, this register is readable/writable.

Address: <ENCA_n_base> + 00_H

Value after reset: 0000_H

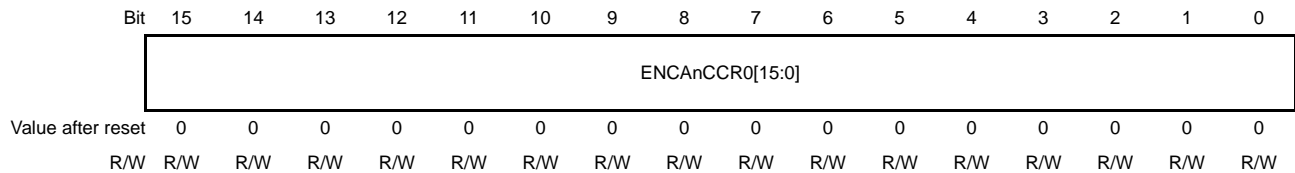


Table 28.15 ENCA_nCCR0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------------------|---|
| 15 to 0 | ENCA _n CCR0 [15:0] | Capture/Compare Register 0 Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCA _n CTL.ENCA _n LDE setting. See the description of the ENCA _n LDE bit in ENCA control register ENCA _n CTL for details. <ul style="list-style-type: none"> • If ENCA_nCTL.ENCA_nCRM0 = 0: ENCA_nCCR0 is compare register. Set the value to be compared with the timer counter value. • If ENCA_nCTL.ENCA_nCRM0 = 1: ENCA_nCCR0 is capture register. The captured timer counter value is stored. |

28.3.8 ENCA_nCCR1 – ENCA Capture/Compare Register 1

This register is a 16-bit capture/compare register 1.

Access: This register can only be read/written in 16-bit units.
When used as a capture register, this register is only readable. Writing to this register is ignored.
When used as a compare register, this register is readable/writable.

Address: <ENCA_n_base> + 04_H

Value after reset: 0000_H

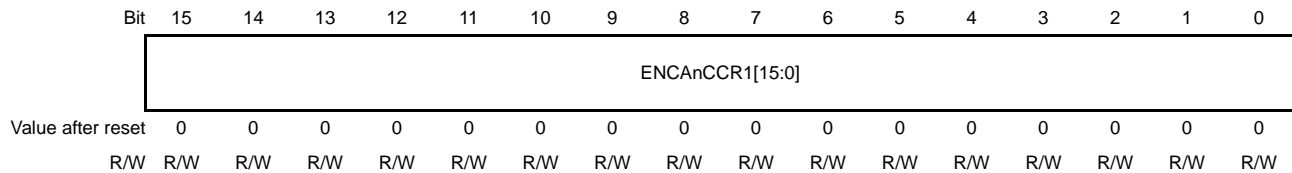


Table 28.16 ENCA_nCCR1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------------|---|
| 15 to 0 | ENCA _n CCR1 [15:0] | Capture/Compare Register 1 During capture operation, the trigger for capturing to this register differs according to the ENCA _n CTL.ENCA _n CTS setting. See the description of the ENCA _n CTS bit in ENCA control register ENCA _n CTL for details. <ul style="list-style-type: none"> If ENCA_nCTL.ENCA_nCRM1 = 0: ENCA_nCCR1 is compare register Set the value to be compared with the timer counter value. If ENCA_nCTL.ENCA_nCRM1 = 1: ENCA_nCCR1 is capture register The captured timer counter value is stored. |

28.3.9 ENCAncNT — ENCA Counter Register

This register is the 16-bit timer counter register.

Access: This register can be read/written in 16-bit units.
This register can be written only when the operation is stopped.

Address: <ENCAnc_base> + 08_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | ENCAncNT[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 28.17 ENCAncNT Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 15 to 0 | ENCAncNT [15:0] | Counter Register <ul style="list-style-type: none"> ENCAncTE.ENCAncTE status: 0 (initial setting): Count stop An arbitrary value can be set to timer counter. ENCAncTE.ENCAncTE status: 0 → 1 (operation start): Count operation start Up/down count operation is started with the set arbitrary value. ENCAncTE.ENCAncTE status: 1 (operating): Counting Up/down count operation is performed. ENCAncTE.ENCAncTE status: 1 → 0 (stopped): Count stop The counter value immediately before the operation was stopped is held, and the count operation is stopped. |

28.3.10 ENCA_nTE — ENCA Timer Enable Status Register

This register indicates the operating status of ENCA_n.

Access: This register can only be read in 8-bit units.

Address: <ENCA_n_base> + 14_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|----------------------|
| | — | — | — | — | — | — | — | ENCA _n TE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 28.18 ENCA_nTE Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ENCA _n TE | <p>Timer Status Enable</p> <p>This is a status bit that indicates the operation enabled/stopped status of ENCA_n.</p> <p>This bit is cleared to 0 when 1 is written to ENCA_nTT.ENCA_nTT.</p> <p>This bit is set to 1 when 1 is written to ENCA_nTS.ENCA_nTS, or when the input signal of ENCA_nTSST is set to High level.</p> <p>0: Operation stopped status</p> <p>1: Operation enabled status</p> |

28.3.11 ENCA_nTS — ENCA Timer Start Trigger Register

This register provides the trigger bit for setting the ENCA_n to the operation enabled state.

Access: This register can only be written in 8-bit units.
This register always returns 00_H when read.

Address: <ENCA_n_base> + 18_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|----------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ENCA _n TS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 28.19 ENCA_nTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ENCA _n TS | Timer Start Trigger This is the trigger bit that sets the ENCA _n to the operation enabled state. 0: Writing is ignored. 1: The ENCA _n is set to the operation enabled state by setting ENCA _n TE.ENCA _n TE = 1. |

28.3.12 ENCA_nTT — ENCA Timer Stop Trigger Register

This register provides the trigger bit for setting the ENCA_n to the operation stopped state.

Access: This register can only be written in 8-bit units.
This register always returns 00_H when read.

Address: <ENCA_n_base> + 1C_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|----------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ENCA _n TT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 28.20 ENCA_nTT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ENCA _n TT | Timer Stop Trigger This is the trigger bit that sets the ENCA _n to the operation stopped state. 0: Writing is ignored. 1: Clears ENCA _n TE. ENCA _n TE to 0, to set the ENCA _n to the count operation stopped state. |

28.4 Operation

The ENCA_n operates the timer counter with counter up/down control and clear control by encoder inputs. The ENCA_nCCR0 and ENCA_nCCR1 registers can be used as dedicated compare registers or as dedicated capture registers.

28.4.1 Timer Counter Operation

The timer counter operations of the ENCA_n are described below.

The figure below shows the operation phases. See the corresponding section with the section number for detailed descriptions on each operation.

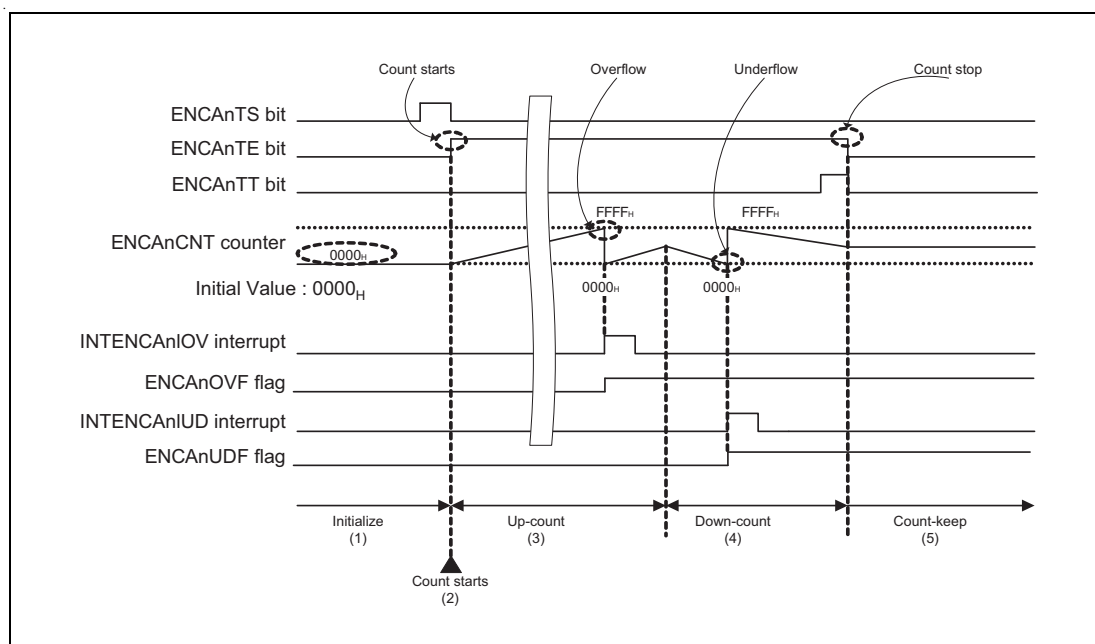


Figure 28.2 Timer Counter Initial Setting Value Setting/Start/Stop

(1) Timer counter initial setting value setting

The initial setting value of the ENCA_n counter register (ENCA_nCNT) can be set in the counter operation stopped status (ENCA_nTE = 0).

(2) Timer counter startup

By writing 1 to the timer start trigger bit (ENCA_nTS), the timer status enable bit (ENCA_nTE) is set to 1, the count operation is enabled, and counting operation is performed upon detection of the valid edge of the encoder input.

(3) Overflow operation

An overflow occurs when up-counting is performed when the counter value is FFFF_H. If the counter value changes from FFFF_H to 0000_H, an overflow interrupt (INTENCA_nIOV) is generated, and the overflow flag (ENCA_nOVF) is set to 1. The overflow flag (ENCA_nOVF) is cleared to 0 when 1 is set to the overflow flag (ENCA_nCLOV). For details about the operation, see **Section 28.6.1, Overflow Occurrence and Overflow Flag Clear Operation.**

(4) Underflow operation

An underflow occurs when down-counting is performed when the counter value is 0000_H. If the counter value changes from 0000_H to FFFF_H, an underflow interrupt (INTENCA_nIUD) is generated, and the underflow flag (ENCA_nUDF) is set to 1. The underflow flag (ENCA_nUDF) is cleared to 0 when 1 is set to the underflow flag (ENCA_nCLUD). For details about the operation, see **Section 28.6.2, Underflow Occurrence and Underflow Flag Clear Operation**.

(5) Timer counter stop

By writing 1 to the timer stop trigger bit (ENCA_nTT), the timer status enable bit (ENCA_nTE) is cleared to 0, and the count operation is stopped. At this time, the timer counter is not reset to 0000_H and holds the value before count operation stop.

28.4.2 Up/Down Control of Timer Counter

Up/down control is performed by judging the phase of the encoder inputs (ENCAnE0, ENCAne1) according to the settings of ENCAAnUDS[1:0].

28.4.2.1 When ENCAAnUDS[1:0] Bits in ENCAAnCTL = 00_B

Table 28.21 When ENCAAnUDS[1:0] = 00_B

| ENCAAnUDS1 | ENCAAnUDS0 | Operation Description | | | |
|------------|------------|-----------------------|----------------|--------------------|----|
| | | ENCAAnE0 input | ENCAAnE1 input | Counting operation | |
| 0 | 0 | Rising edge | High level | Down | |
| | | Falling edge | | | |
| | | Both edges | | | |
| | | Rising edge | Low level | | Up |
| | | Falling edge | | | |
| | | Both edges | | | |

The valid edge for ENCAAnE0 is specified by setting ENCAAnEIS[1:0].

“Counting operation” indicates that counting up or down proceeds when the valid edges of the ENCAAnE0 signal matches the valid level ENCAAnE1 signal.

The following timing chart shows counting operation when ENCAAnUDS[1:0] = 00_B.

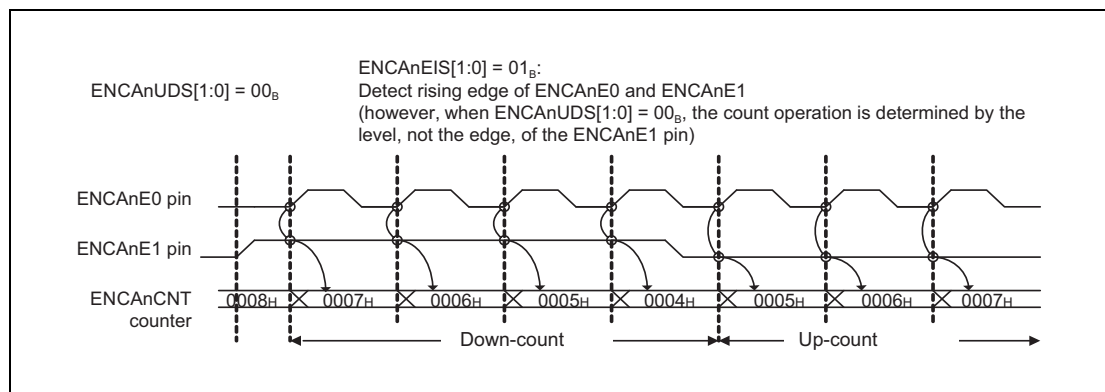


Figure 28.3 Counting Operation when ENCAAnUDS[1:0] in ENCAAnCTL = 00_B

28.4.2.2 When ENCA_nUDS[1:0] Bits in ENCA_nCTL = 01_B

Table 28.22 When ENCA_nUDS[1:0] = 01_B

| ENCA _n UDS1 | ENCA _n UDS0 | Operation Description | | | | |
|------------------------|------------------------|----------------------------|----------------------------|--------------------|----|------|
| | | ENCA _n E0 input | ENCA _n E1 input | Counting operation | | |
| 0 | 1 | Low level | Rising edge | Down | | |
| | | | Falling edge | | | |
| | | | Both edges | | | |
| | | High level | Rising edge | | | |
| | | | Falling edge | | | |
| | | | Both edges | | | |
| | | Rising edge | Low level | Up | | |
| | | Falling edge | | | | |
| | | Both edges | | | | |
| | | Rising edge | High level | | Up | |
| | | Falling edge | | | | |
| | | Both edges | | | | |
| | | Simultaneous input | | | | Hold |

The valid edges for ENCA_nE0 and ENCA_nE1 are specified by setting ENCA_nEIS[1:0].

“Counting operation” indicates that counting up or down proceeds when the valid edges and levels of the signals on ENCA_nE0 and ENCA_nE1 match.

The following timing chart shows counting operation when ENCA_nUDS[1:0] = 01_B.

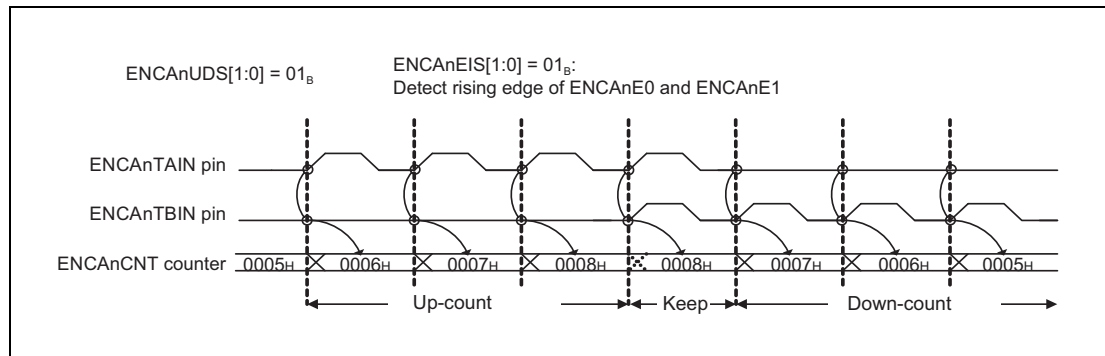


Figure 28.4 Counting Operation when ENCA_nUDS[1:0] in ENCA_nCTL = 01_B

28.4.2.3 When ENCA_nUDS[1:0] Bits in ENCA_nCTL = 10_B

Table 28.23 When ENCA_nUDS[1:0] = 10_B

| ENCA _n UDS1 | ENCA _n UDS0 | Operation Description | | |
|------------------------|------------------------|----------------------------|----------------------------|--------------------|
| | | ENCA _n E0 input | ENCA _n E1 input | Counting operation |
| 1 | 0 | Rising edge | Low level | Down |
| | | Rising edge | Falling edge | |
| | | Falling edge | Low level | Up |
| | | Falling edge | Falling edge | |
| | | Low level | Rising edge | Hold |
| | | Rising edge | Rising edge | |
| | | High level | Rising edge | |
| | | Falling edge | Rising edge | |
| | | Low level | Falling edge | |
| | | Rising edge | High level | |
| | | High level | Falling edge | |
| | | Falling edge | High level | |

Valid edge specification for ENCA_nE0 and ENCA_nE1 (setting to ENCA_nEIS[1:0]) is invalid.

The following timing chart shows counting operation when ENCA_nUDS[1:0] = 10_B.

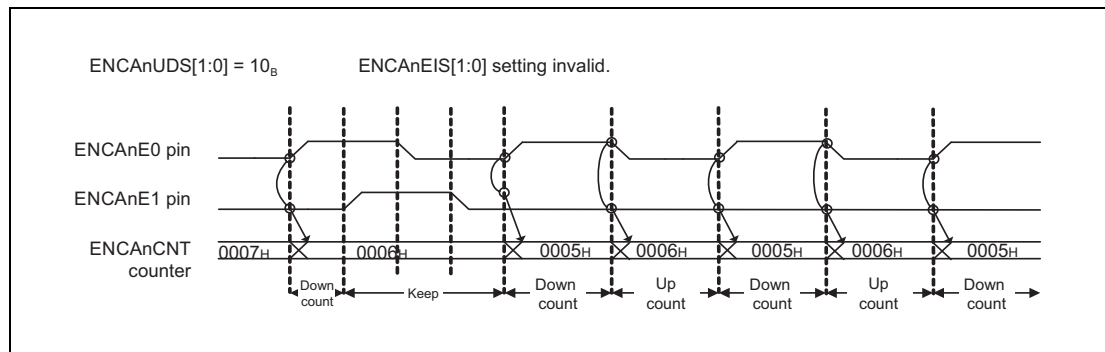


Figure 28.5 Counting Operation when ENCA_nUDS[1:0] in ENCA_nCTL = 10_B

28.4.2.4 When ENCA_nUDS[1:0] Bits in ENCA_nCTL = 11_B

Table 28.24 When ENCA_nUDS[1:0] = 11_B

| ENCA _n UDS1 | ENCA _n UDS0 | Operation Description | | | |
|------------------------|------------------------|----------------------------|----------------------------|--------------------|------|
| | | ENCA _n E0 input | ENCA _n E1 input | Counting operation | |
| 1 | 1 | Low level | Falling edge | Down | |
| | | Rising edge | Low level | | |
| | | High level | Rising edge | | |
| | | Falling edge | High level | | |
| | | Rising edge | High level | Up | |
| | | High level | Falling edge | | |
| | | Falling edge | Low level | | |
| | | Low level | Rising edge | | |
| | | Simultaneous input | | | Hold |

Valid edge specification for ENCA_nE0 and ENCA_nE1 (settings of ENCA_nEIS[1:0]) is invalid.

When the valid edges of the signals on ENCA_nE0 and ENCA_nE1 coincide, the count is retained.

The following timing chart shows counting operation when ENCA_nUDS[1:0] = 11_B

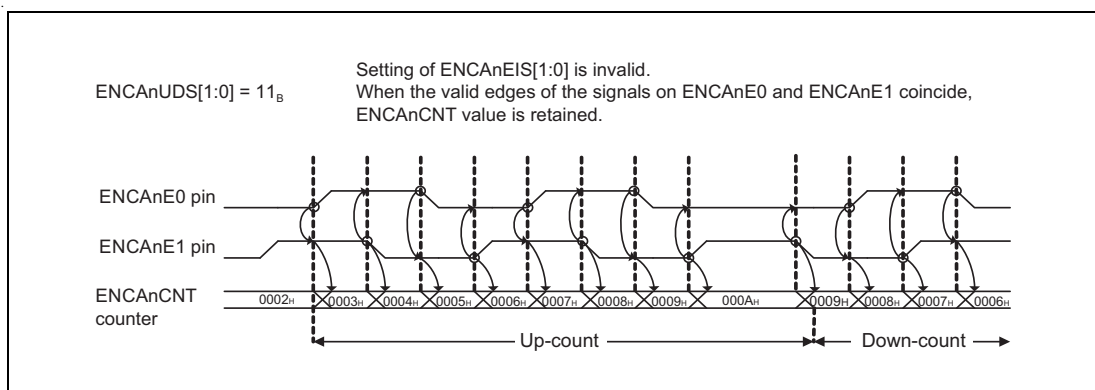


Figure 28.6 Counting Operation when ENCA_nUDS[1:0] = 11_B

28.4.3 Timer Counter Clear Control by Encoder Input

The following conditions lead to clearing of the timer counter to 0000_H.

- Detection of a valid edge of the encoder clearing input signal (signal on the ENCA_nEC pin)
- Level detection of the encoder input and encoder clearing input signals (signals on the ENCA_nE0, ENCA_nE1, and ENCA_nEC pins)

Two types of clearing methods can be selected by controlling the ENCA_nSCE, ENCA_nZCL, ENCA_nBCL, ENCA_nACL, ENCA_nECS[1:0] bits of the ENCA_nIOC1 register.

Table 28.25 Timer Counter Clear Control by Encoder Input

| Clearing method | ENCA _n SCE | ENCA _n ZCL | ENCA _n BCL | ENCA _n ACL | ENCA _n ECS[1:0] |
|-----------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------|
| (1) | 0 | Invalid | Invalid | Invalid | Valid |
| (2) | 1 | Valid | Valid | Valid | Invalid |

28.4.3.1 Clearing Method when ENCA_nSCE = 0

- Upon detection of the valid edge of ENCA_nEC, the timer counter is cleared to 0000_H in synchronization with the operation clock.
- The valid edge of ENCA_nEC is specified by the setting of the ENCA_nECS[1:0] bits.
- The settings of the ENCA_nZCL, ENCA_nBCL, and ENCA_nACL bits are invalid.
- An encoder clear interrupt request signal (INTENCA_nIEC) is output simultaneously with timer counter clearing.

For details about clear operation when ENCA_nSCE = 0, see the timing chart in **Section 28.6.19, Capture Operation Performed upon Clearing by ENCA_nEC when ENCA_nSCE = 0.**

28.4.3.2 Clearing Method when ENCA_nSCE = 1

- When the clear levels of the ENCA_nEC, ENCA_nE1, and ENCA_nE0 inputs are detected, the timer counter is cleared to 0000_H in synchronization with the operating clock.
- Specify the clear levels of the ENCA_nEC, ENCA_nE1, and ENCA_nE0 inputs by the settings of the ENCA_nZCL, ENCA_nBCL, and ENCA_nACL bits.
- The settings of the ENCA_nECS[1:0] bits are invalid.
- An encoder clear interrupt request signal (INTENCA_nIEC) is output simultaneously with timer counter clearing.

The clearing conditions of the timer counter according to the ENCA_nZCL, ENCA_nBCL, and ENCA_nACL settings are listed in the table below.

Table 28.26 Clearing Conditions of Timer Counter

| Counter Clear Condition Setting | | | Encoder Input Level | | |
|---------------------------------|-----------------------|-----------------------|----------------------|----------------------|----------------------|
| ENCA _n ZCL | ENCA _n BCL | ENCA _n ACL | ENCA _n EC | ENCA _n E1 | ENCA _n E0 |
| 0 | 0 | 0 | Low | Low | Low |
| 0 | 0 | 1 | Low | Low | High |
| 0 | 1 | 0 | Low | High | Low |
| 0 | 1 | 1 | Low | High | High |
| 1 | 0 | 0 | High | Low | Low |
| 1 | 0 | 1 | High | Low | High |
| 1 | 1 | 0 | High | High | Low |
| 1 | 1 | 1 | High | High | High |

28.4.4 Functions of ENCA_nCCR0

28.4.4.1 Compare Function

- When ENCA_nCRM0 = 0, the ENCA_nCCR0 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCA_nCCR0 setting value, a compare 0 match interrupt (INTENCA_nI0) is output.
- When ENCA_nECM0 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is up-count.

Table 28.27 Compare Function of ENCA_nCCR0

| ENCA _n CCR0 function | Compare match clear control | Next count operation | Timer counter clearing upon compare match with ENCA _n CCR0 |
|---------------------------------|-----------------------------|----------------------|---|
| ENCA _n CRM0 | ENCA _n ECM0 | | |
| 0 (Compare) | 0 | Up-count | Does not clear (continues count operation). |
| | | Down-count | |
| | 1 | Up-count | Clears timer counter to 0000 _H . |
| | | Down-count | Does not clear (continues count operation). |

When ENCA_nLDE = 1

- Upon occurrence of an underflow, the setting value of the ENCA_nCCR0 register is loaded to the timer counter.
- An underflow interrupt (INTENCA_nIUD) is output.

NOTE

For the timing chart when ENCA_nLDE = 1, see the timing charts in **Section 28.6.8, Using the ENCA_nLDE Function Immediately after Startup** and **Section 28.6.12, Up-count after Conflict between ENCA_nLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

28.4.4.2 Capture Function

- When ENCA_nCRM0 = 1, the ENCA_nCCR0 register functions as a dedicated capture register.
- Upon valid edge detection of the capture trigger input 0 (ENCA_nTTIN0), the value of the timer counter is stored into ENCA_nCCR0.
- A capture 0 interrupt (INTENCA_nI0) is output during capture operation.

NOTE

For details about capture operation to ENCA_nCCR0, see the timing charts in **Section 28.6.14, Capture Operation between Count Clocks (ENCA_nCCR0)** and **Section 28.6.17, Encoder operation when compare match clear control is disabled**.

28.4.5 Functions of ENCA_nCCR1

28.4.5.1 Compare Function

- When ENCA_nCRM1 = 0, the ENCA_nCCR1 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCA_nCCR1 setting value, a compare 1 match interrupt (INTENCA_n1) is output.
- When ENCA_nECM1 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is down-count.

Table 28.28 Compare Function of ENCA_nCCR1

| ENCA _n CCR1 function | Compare match clear control | Next count operation | Timer counter clearing upon compare match with ENCA _n CCR1. |
|---------------------------------|-----------------------------|----------------------|--|
| ENCA _n CRM1 | ENCA _n ECM1 | | |
| 0 (Compare) | 0 | Up-count | Does not clear (continues count operation). |
| | | Down-count | |
| | 1 | Up-count | Does not clear (continues count operation). |
| | | Down-count | Clears timer counter to 0000 _H . |

Compare match interrupt detection mask function

- When $ENCA_nCME = 1$, the compare 1 match interrupt detection mask function is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the $ENCA_nCCR1$ setting value, and interrupts are then masked for the second and subsequent compare matches.
- When $ENCA_nMCS = 0$, the compare 1 match interrupt detection mask function is disabled by a write operation to the $ENCA_nCCR1$ register.
- When $ENCA_nMCS = 1$, the compare 1 match interrupt detection mask function is disabled by a timer counter clear operation accompanying phase Z or by a timer counter clear operation upon match between the $ENCA_nCCR0$ register value and the timer counter value.
- When $ENCA_nMCS = 1$ and $ENCA_nLDE = 1$, the compare 1 match interrupt detection mask function is disabled by a loading operation of the $ENCA_nCCR0$ register to the timer counter upon underflow detection.
- Setting $ENCA_nECM1$ to 1 is prohibited when enabling the compare 1 match interrupt detection mask function.

Table 28.29 Compare Match Interrupt Detection Mask Function of $ENCA_nCCR1$

| $ENCA_nCCR1$ function | Compare 1 match interrupt mask | Interrupt mask cancel trigger | | Compare 1 match interrupt output upon compare match with $ENCA_nCCR1$ |
|--------------------------------------|--------------------------------|-------------------------------|---|---|
| $ENCA_nCRM1$ | $ENCA_nCME$ | $ENCA_nMCS$ | Underflow occurrence upon $ENCA_nLDE = 1$ | |
| 0 (Compare) | 0 (Mask function disabled) | (Setting invalid) | — | Outputs compare 1 match interrupt upon each compare match. |
| | 1 (Mask function enabled) | | 0 (Write operation to $ENCA_nCCR1$) | Occurred (Loading of $ENCA_nCCR0$ to timer counter) |
| 1 (Timer counter clear operation) | | | | |

28.4.5.2 Capture Function

When ENCA_nCRM1 = 1, the ENCA_nCCR1 register functions as a dedicated capture register.

NOTE

For details about capture operation to ENCA_nCCR1, see the timing chart in **Section 28.6.13, Capture Operation between Count Clocks (ENCA_nCCR1)**.

The operations for each of the ENCA_nCTS settings are shown in the table below.

Table 28.30 Operations for each of the ENCA_nCTS settings

| ENCA _n CCR1 function | Capture trigger selection | Capture trigger signal | Timer counter clearing | Interrupt occurrence |
|---------------------------------|---------------------------|--|-------------------------------|---|
| ENCA _n CRM1 | ENCA _n CTS | | | |
| 1 (Capture) | 0 | Capture trigger 1 input (ENCA _n TTIN1) | Does not clear timer counter. | (1) Capture 1 interrupt (INTENCA _n I1) |
| | 1 | Encoder clear input (specified by ENCA _n SCE) | Clears timer counter. | (1) Capture 1 interrupt (INTENCA _n I1) (2) Encoder clear interrupt (INTENCA _n IEC) |

NOTE

For details about the timing chart when ENCA_nCTS = 0 or ENCA_nCTS = 1, see the following: **Section 28.6.3, Count Clear and Capture Operation by Encoder Clear Input (ENCA_nEC pin)**, **Section 28.6.4, Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)**, **Section 28.6.5, Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCA_nEC Pin)**, **Section 28.6.11, Conflict between ENCA_nLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input** and **Section 28.6.12, Up-count after Conflict between ENCA_nLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

28.4.5.3 Timer Counter Clearing upon Compare Register Match

Timer counter clearing upon compare match between the value of the timer counter and the ENCA_nCCR0/ENCA_nCCR1 setting value, according to the settings of the ENCA_nECM1 and ENCA_nECM0 bits in the ENCA_nCTL register, is detailed in the following table.

Table 28.31 Timer Counter Clearing Operation upon Compare Register Match

| ENCA _n ECM1 and ENCA _n ECM0 | Next count operation | Timer counter clearing upon compare match with ENCA _n CCR1 | Timer counter clearing upon compare match with ENCA _n CCR0 |
|---|-------------------------|--|--|
| 00 | Up-count | Does not clear (continues count operation). | Does not clear (continues count operation). |
| | Down-count | Does not clear (continues count operation). | Does not clear (continues count operation). |
| 01 | Up-count | Does not clear (continues count operation). | Clears timer counter to 0000 _H . |
| | Down-count | Does not clear (continues count operation). | Does not clear (continues count operation). |
| 10 | Up-count | Does not clear (continues count operation). | Does not clear (continues count operation). |
| | Down-count | Clears timer counter to 0000 _H . | Does not clear (continues count operation). |
| 11 | Up-count | Does not clear (continues count operation). | Clears timer counter to 0000 _H . |
| | Down-count | Clears timer counter to 0000 _H . | Does not clear (continues count operation). |

28.4.6 Startup/Stop of Timer Counter

28.4.6.1 Startup of Timer

The timer operation can be started by setting the ENCA_nTS bit to 1. Simultaneous start with other timer can be possible by setting the PIC. For details, see **Section 29.2.3.1, Simultaneous Start Trigger Function**

28.4.6.2 Stop of Timer

When the ENCA_nTT bit is set to 1, the ENCA_nTE bit becomes 0 and the timer stops.

28.5 ENCA Setting Sequences

28.5.1 Encoder timer setting procedure

The encoder timer setting procedure is described below.

Table 28.32 Encoder Timer Setting Procedure

| | Action | Setting status |
|------------------------|---|--|
| Initial Setting | Reset deassertion | Power-on status, operation stopped status (Writing to each register is enabled) |
| ENCAn initial Settings | Perform the following initial settings. <ul style="list-style-type: none"> Setting for counter Setting for counter clear Setting for ENCAAnCCR0 register Setting for ENCAAnCCR1 register | This is the count operation stopped status. The value of the ENCAAnTE bit indicating the operating status is 0. |
| | Perform the counter initial value settings. <ul style="list-style-type: none"> Set any 16-bit value to ENCAAnCNT register. (When, after setting this register, the ENCAAnTS bit is set to 1, the counter operation starts from the set count value.) | The set value is set as the initial value of the counter register value. |
| Operation Start | Perform the counter operation start setting. <ul style="list-style-type: none"> Set the ENCAAnTS bit to 1. | This is the counter operation start status. The value of the ENCAAnTE bit indicating the operating status is 1, and the count clock is supplied to the internal circuit. |
| Operating | Only those registers whose setting can be changed during operation can be rewritten. <ul style="list-style-type: none"> ENCAAnCCR0 register setting ENCAAnCCR1 register setting ENCAAnIOC0 register setting | The count operation set with the initial setting is performed, and up/down counting is performed according to the ENCAAnE0 and ENCAAnE1 pins. |
| Operation Stop | Perform the counter operation stop setting during operation. <ul style="list-style-type: none"> Set the ENCAAnTT bit to 1. | This is the counter operation stopped status. The value of the ENCAAnTE bit indicating the operating status is 0. |
| ENCA stop | Reset | The setting registers are initialized. |

28.5.1.1 Initial Setting Procedure for Counter

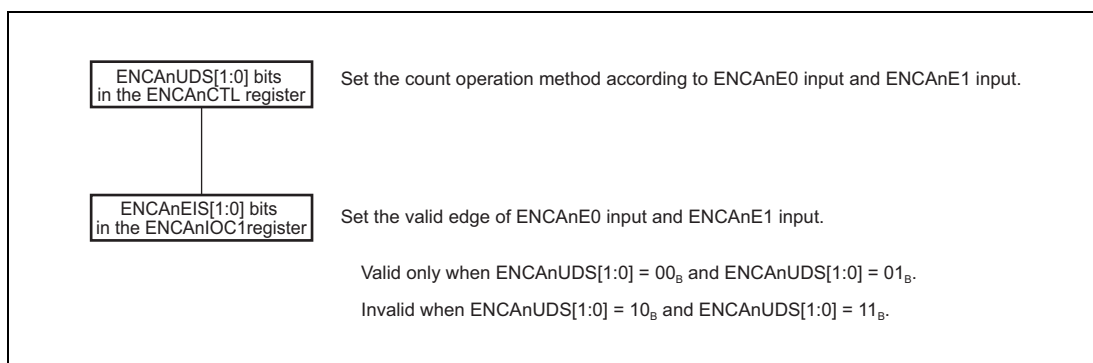


Figure 28.7 Initial Setting Procedure for Counter

28.5.1.2 Initial Setting Procedure for Counter Clear

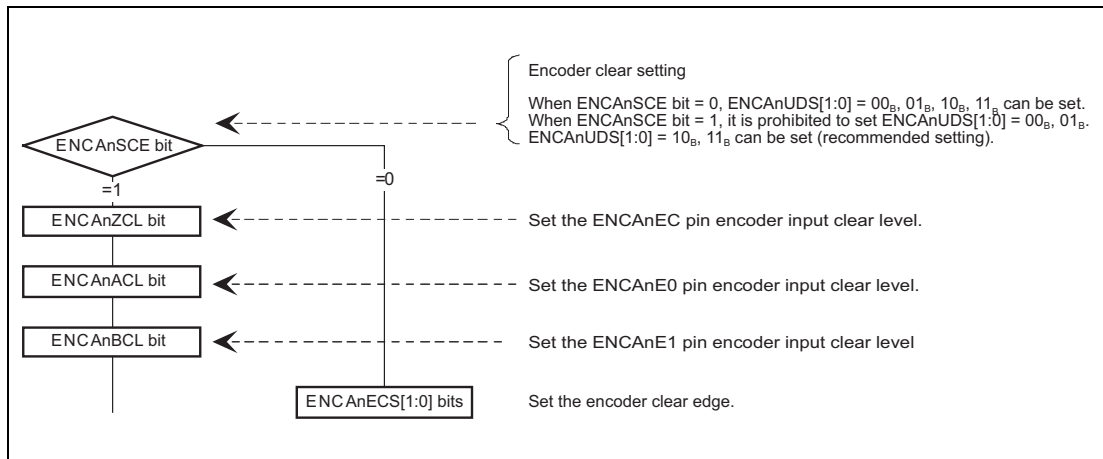


Figure 28.8 Initial Setting Procedure for Counter Clear

28.5.1.3 Setting Procedure for ENCAAnCCR0 Register

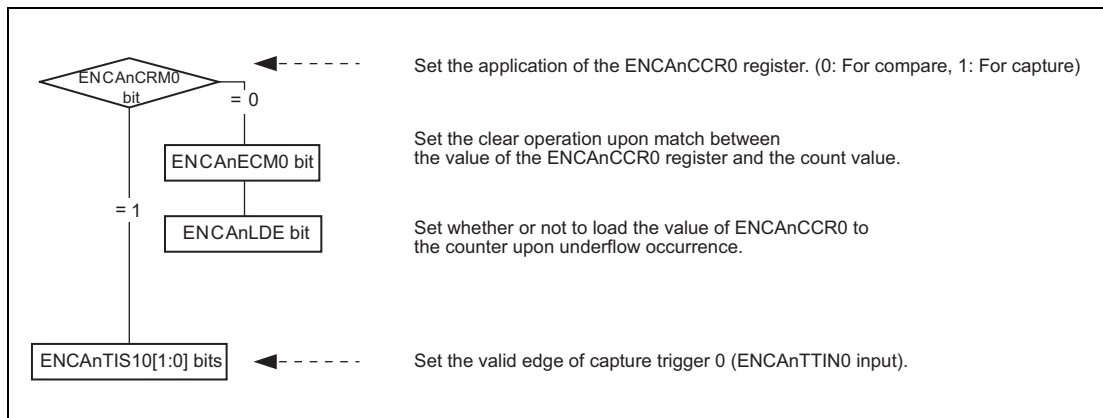


Figure 28.9 Setting Procedure for ENCAAnCCR0 register

28.5.1.4 Setting Procedure for ENCA_nCCR1 Register

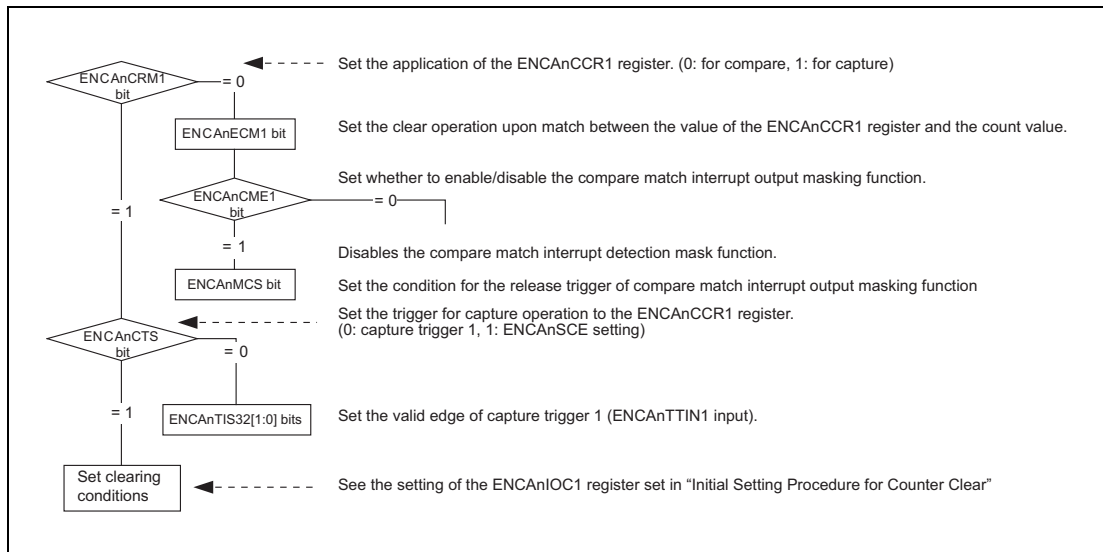


Figure 28.10 Setting Procedure for ENCA_nCCR1 register

28.6 Timing Chart

28.6.1 Overflow Occurrence and Overflow Flag Clear Operation

When an up-counting is performed while the counter value is $FFFF_H$, an overflow occurs. If an overflow occurs, an overflow interrupt (INTENCAnIOV) is output and the overflow flag (ENCAnOVF) is set to 1. The overflow flag clear bit (ENCAnCLOV) is set to 1, the overflow flag (ENCAnOVF) is cleared to 0.

The overflow occurrence and overflow flag clear operation are described as follows.

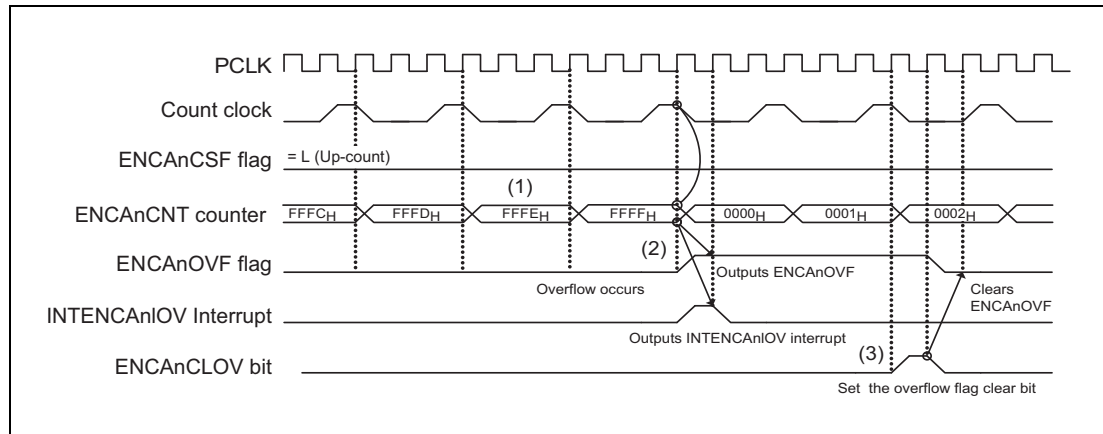


Figure 28.11 Overflow Occurrence and Overflow Flag Clear Setting

- (1) The count value is up-counted from $FFFE_H$ to $FFFF_H$.
- (2) When the count value changes from $FFFF_H$ to 0000_H , an overflow occurs. At the same time, an overflow interrupt is output, and the overflow flag is set to 1.
- (3) An overflow flag is cleared to 0 by setting 1 to the ENCAncCLOV bit in the ENCAncFGC register according to the clearing procedure. In addition, an overflow flag is also cleared by setting the ENCAncTS bit in the ENCAncTS register to 1 while ENCAncTE.ENCAncTE is 0, or by setting an input signal of the ENCAncTSST (simultaneous start trigger input) to high.

28.6.2 Underflow Occurrence and Underflow Flag Clear Operation

When down-counting is performed while the counter value is 0000_H, an underflow occurs. If an underflow occurs, an underflow interrupt (INTENCAnUD) is output and the underflow flag (ENCAnUDF) is set to 1. The underflow flag clear bit (ENCAnCLUD) is set to 1, the underflow flag (ENCAnUDF) is cleared to 0.

The underflow occurrence and underflow flag clear operation are described as follows.

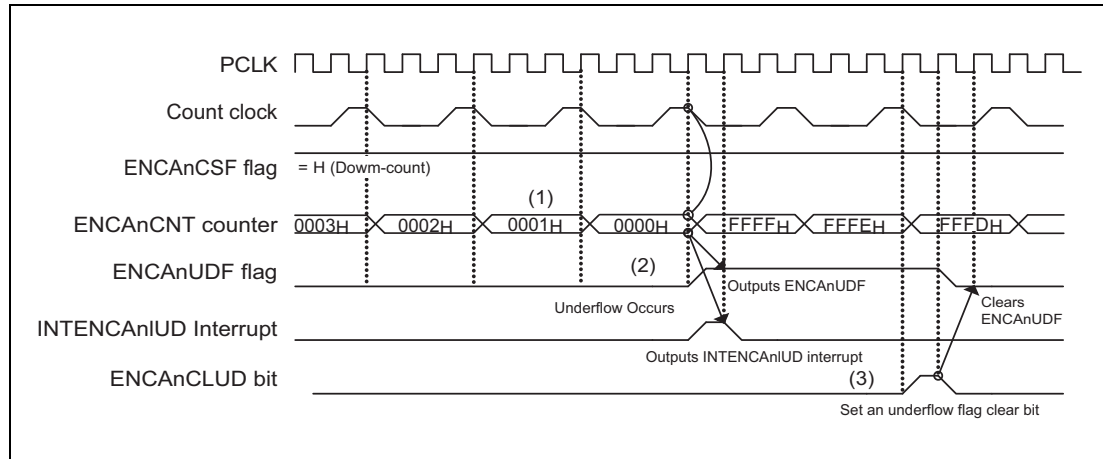


Figure 28.12 Underflow Occurrence and Underflow Flag Clear Setting

- (1) The count value is down-counted from 0001_H to 0000_H.
- (2) When the count value changes from 0000_H to FFFF_H, an underflow occurs. At the same time, an underflow interrupt is output, and the underflow flag is set to 1.
- (3) An underflow flag is cleared to 0 by setting 1 to the ENCAncCLUD bit in the ENCAncFGC register according to the clearing procedure. In addition, an underflow flag is also cleared by setting the ENCAncTS bit in the ENCAncTS to 1 while ENCAncTE.ENCAncTE is 0, or by setting an input signal of the ENCAncTSST (simultaneous start trigger input) to high.

28.6.3 Count Clear and Capture Operation by Encoder Clear Input (ENCAnEC pin)

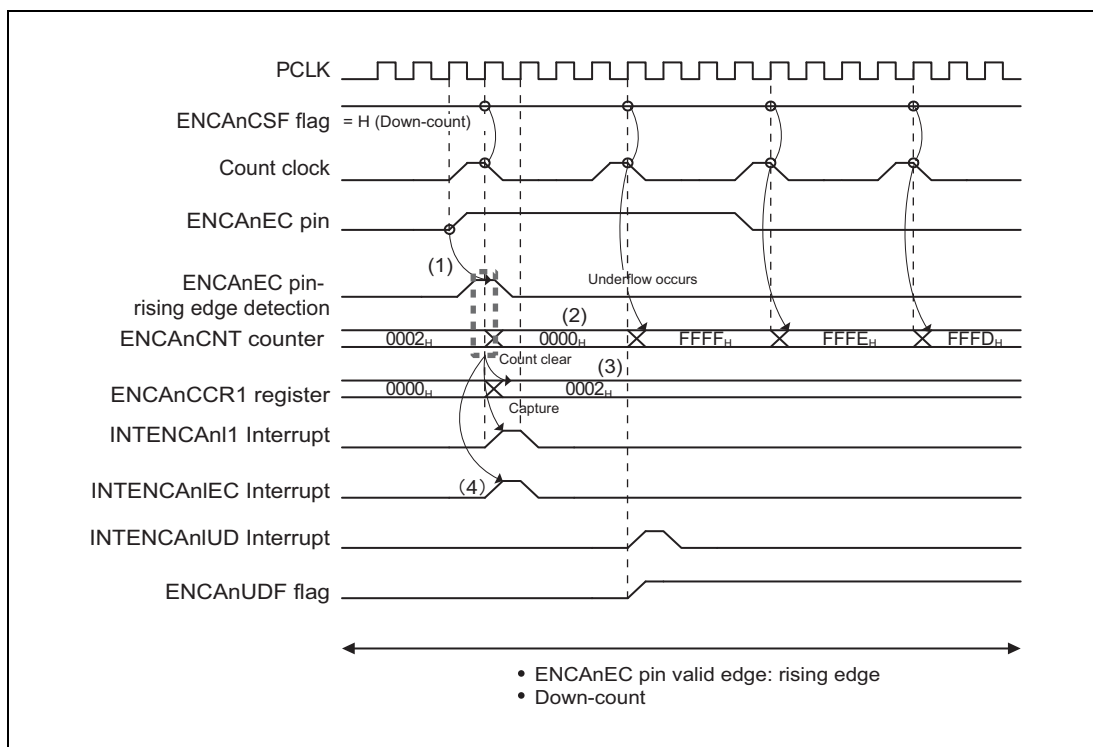


Figure 28.13 Timing Diagram of Count Clear and Capture Operation by Encoder Clear Input (ENCAnEC pin)

[Setting Conditions]

- ENCAAnCTL.ENCAAnCRM1 = 1
(ENCAAnCCR1 register is selected as capture)
- ENCAAnCTL.ENCAAnCTS = 1
(ENCAAnEC pin input is selected as a capture trigger input)
- ENCAAnIOC1.ENCAAnECS[1:0] bits = 01_B
(Selected as a rising edge detection of the ENCAAnEC pin input)

- (1) Capture operation is performed by the rising edge of the ENCAAnEC pin trigger.
- (2) Clear operation is performed by an input to the ENCAAnEC pin and the count value is reset to 0000_H.
- (3) The counter value (0002_H) is captured to the ENCAAnCCR1 at the rising edge of the ENCAAnEC pin.
- (4) At the same time, by the ENCAAnEC pin input, a clear interrupt (INTENCAAnIEC) and capture interrupt (INTENCAAn1) are output.

28.6.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

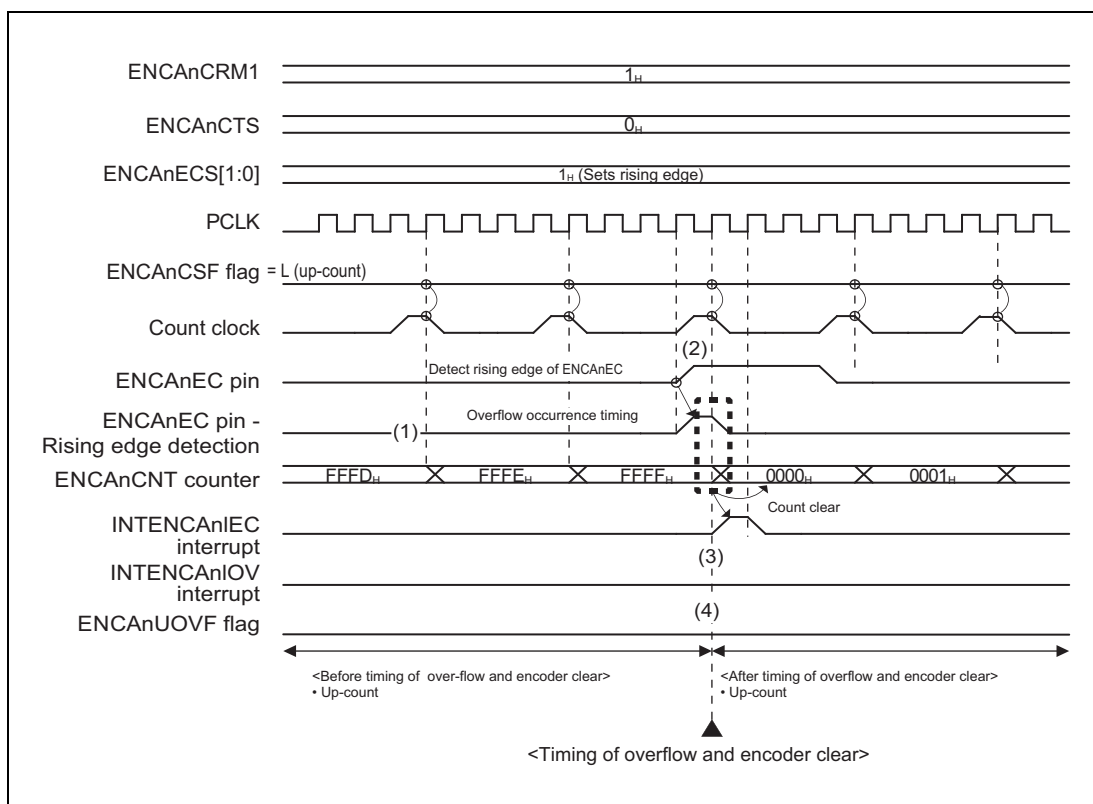


Figure 28.14 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC pin)

- (1) An up-count from FFFD_H is continuously performed.
- (2) When an overflow occurs if the count value is FFFF_H, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. The counter value is cleared to 0000_H.
- (3) When the counter value is cleared by the encoder clear input, a clear interrupt (INTENCAnIEC) by encoder clear input is output simultaneously. Because a clear operation by the phase Z input is performed simultaneously with the overflow occurrence, an overflow interrupt is not output (An overflow does not occur. Clear operation is performed by the phase Z input).
- (4) Because an overflow does not occur as is the case with step 3, the overflow flag is not set.

28.6.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

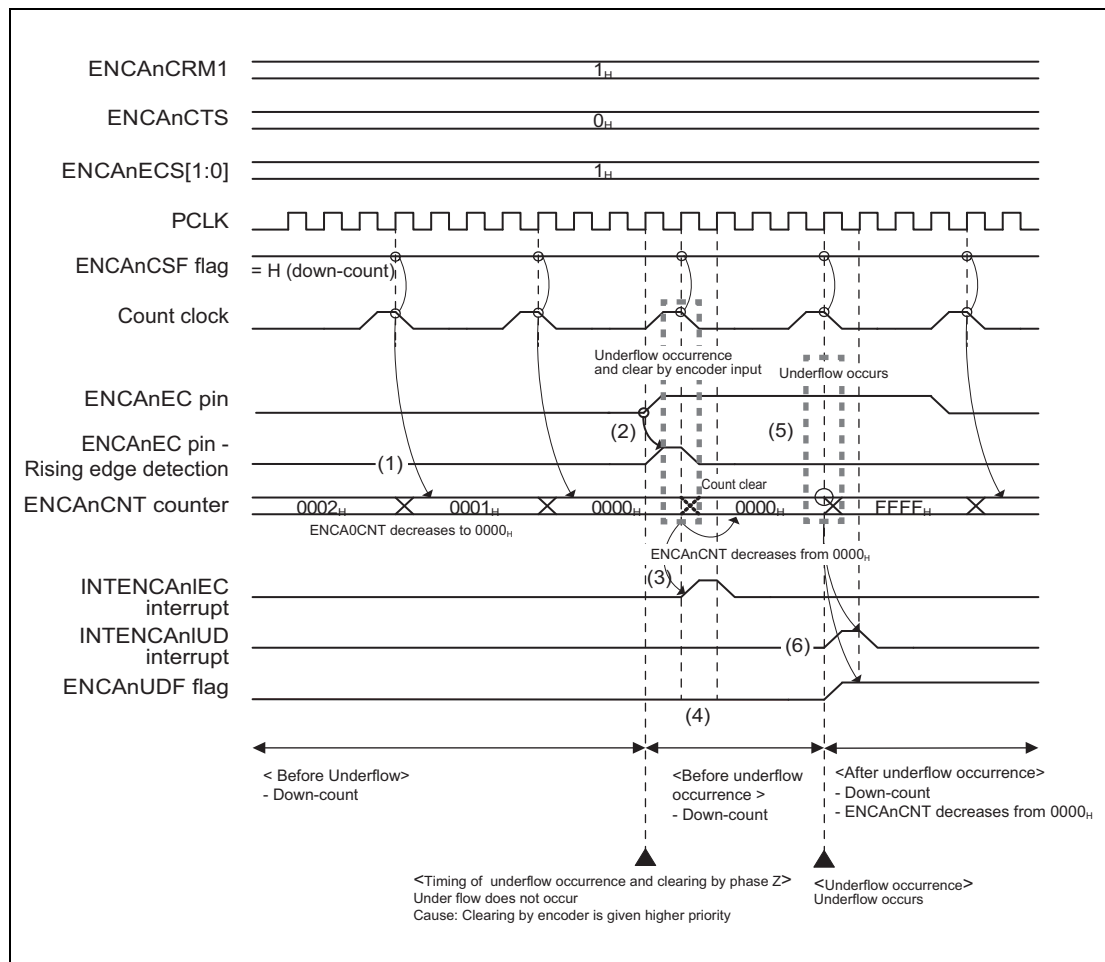


Figure 28.15 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

- (1) A down-count from 0002_H is continuously performed.
- (2) When an underflow occurs if the count value is 0000_H, and the rising edge of ENCAAnEC is detected simultaneously, clear operation by the encoder clear input is performed. Even if the next clock signal is input during clear operation, the counter value remains at 0000_H.
- (3) When the counter value is cleared by the encoder clear input, an encoder clear interrupt (INTENCAAnIEC) is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the underflow occurrence, an underflow interrupt is not output (An underflow does not occur. Clear operation is performed by the encoder clear input).
- (4) Because an underflow does not occur as is the case with step 3, the underflow flag is not set.
- (5) When a further down-count is performed after the counter value changes to 0000_H by clear operation by the encoder clear input, the counter value changes from 0000_H to FFFF_H, and an underflow occurs.
- (6) When an underflow occurs, an underflow interrupt (INTENCAAnIUD) is output, and the underflow flag (ENCAAnUDF) is set.

28.6.6 Overflow Operation Immediately after Startup

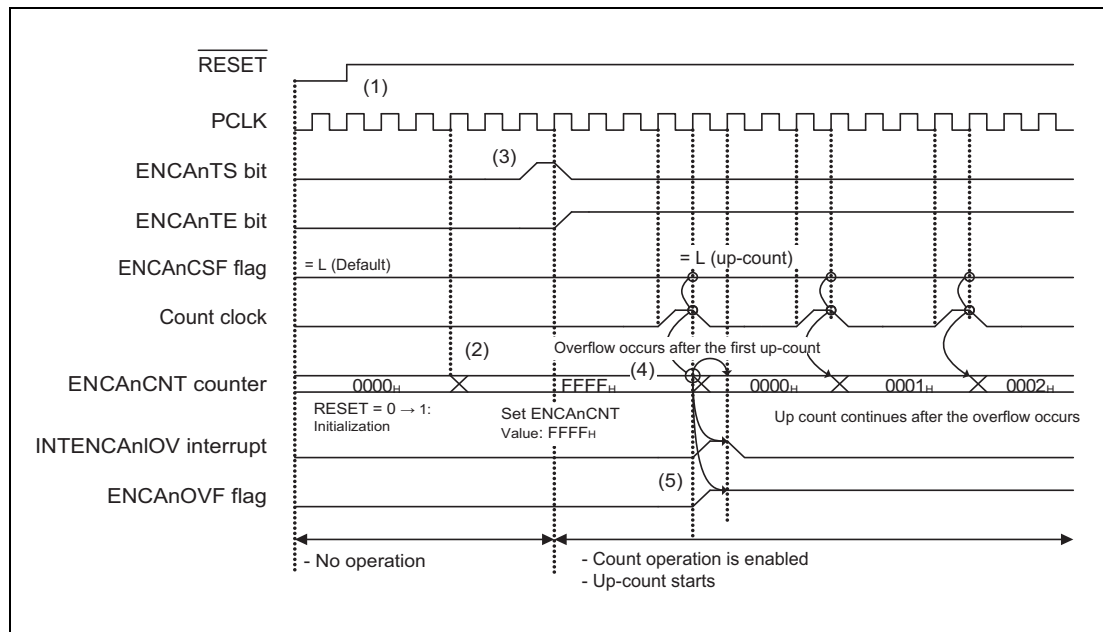


Figure 28.16 Overflow Operation Immediately after Startup

- (1) When the $\overline{\text{RESET}}$ value changes from 0 to 1, the status changes from “reset” to “reset release”.
- (2) The timer counter is set to FFFF_{H} as the initial value.
- (3) ENCA nTS is set to 1, and operation starts. ENCA nTE changes to 1, which indicates that operation is enabled.
- (4) When an up-count is performed from FFFF_{H} which is the initially set count value, the counter value changes from FFFF_{H} to 0000_{H} , and an overflow occurs immediately after operation starts.
- (5) At the same time, by an overflow occurrence immediately after operation starts, an overflow interrupt (INTENCA nIOV) is output, and the overflow flag (ENCA nOVF) is set.

28.6.7 Underflow Operation Immediately after Startup

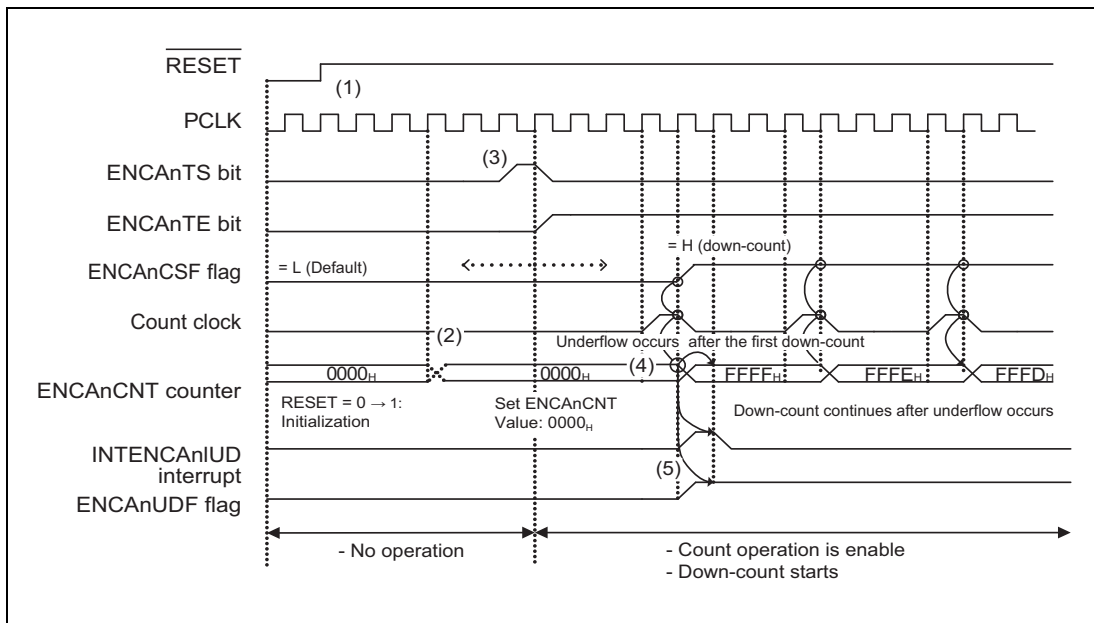


Figure 28.17 Underflow Operation Immediately after Startup

- (1) When the $\overline{\text{RESET}}$ value changes from 0 to 1, the status changes from “reset” to “reset release”.
- (2) The timer counter is set to 0000_H as the initial value.
- (3) ENCAAnTS is set to 1, and operation starts. ENCAAnTE changes to 1, which indicates that operation is enabled.
- (4) When a down-count is performed from 0000_H which is the initially set count value, the counter value changes from 0000_H to FFFF_H, and an underflow occurs immediately after operation starts.
- (5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (INTENCAAnUD) is output, and the underflow flag (ENCAAnUDF) is set.

28.6.8 Using the ENCA_nLDE Function Immediately after Startup

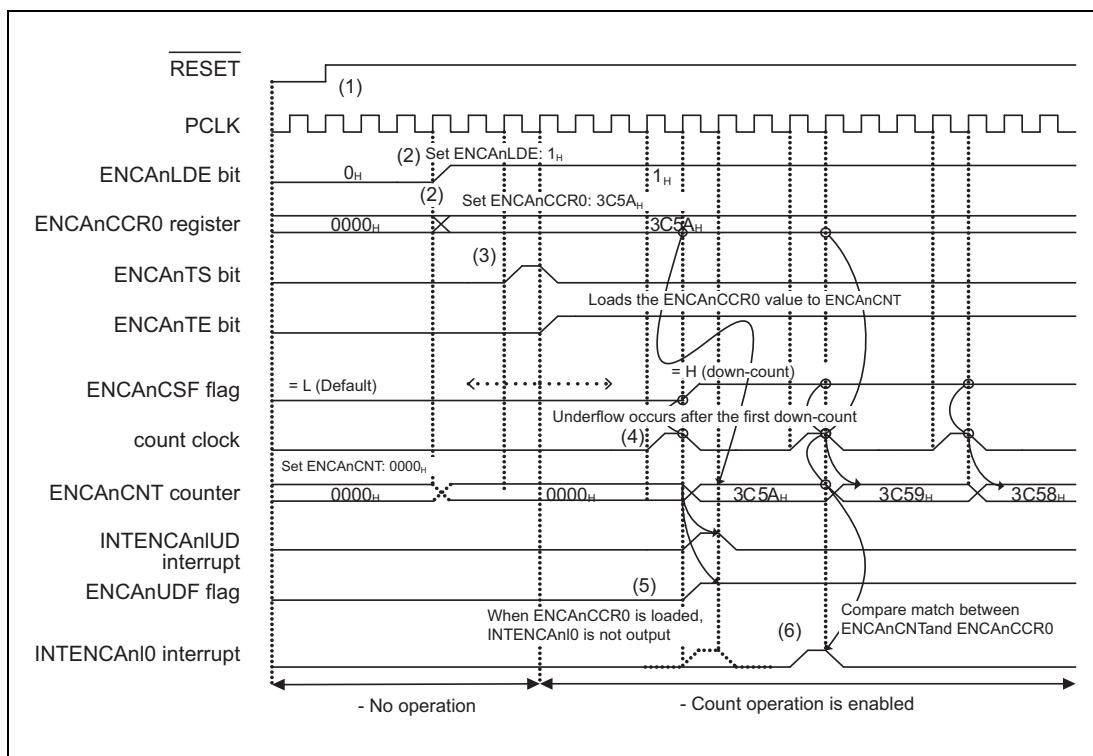


Figure 28.18 Using the ENCA_nLDE Function Immediately after Startup

- (1) When the $\overline{\text{RESET}}$ value changes from 0 to 1, the status changes from “reset” to “reset release”.
- (2) The load enable bit (ENCA_nLDE) is set to 1, capture/compare register 0 (ENCA_nCCR0) is set to 3C5A_H, and the timer counter is set to the initial value 0000_H.
- (3) ENCA_nTS is set to 1, and operation starts. ENCA_nTE changes to 1, which indicates that operation is enabled.
- (4) When a down-count is performed from 0000_H which is the initially set count value, an underflow occurs immediately after operation starts. Because ENCA_nLDE is set to 1, the ENCA_nCCR0 value, 3C5A_H, is loaded to the timer counter (INTENCA_nI0 is not output during loading).
- (5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (INTENCA_nUD) is output, and the underflow flag (ENCA_nUDF) is set (after an underflow occurs, down-count operation from the loaded value (3C5A_H) continues).
- (6) After the ENCA_nCCR0 value is loaded to ENCA_nCNT, a match with ENCA_nCCR0 is detected, and INTENCA_nI0 is output.

28.6.9 ENCA_nLDE Function (Loading Count Value)

(1) <When ENCA_nLDE = 0>

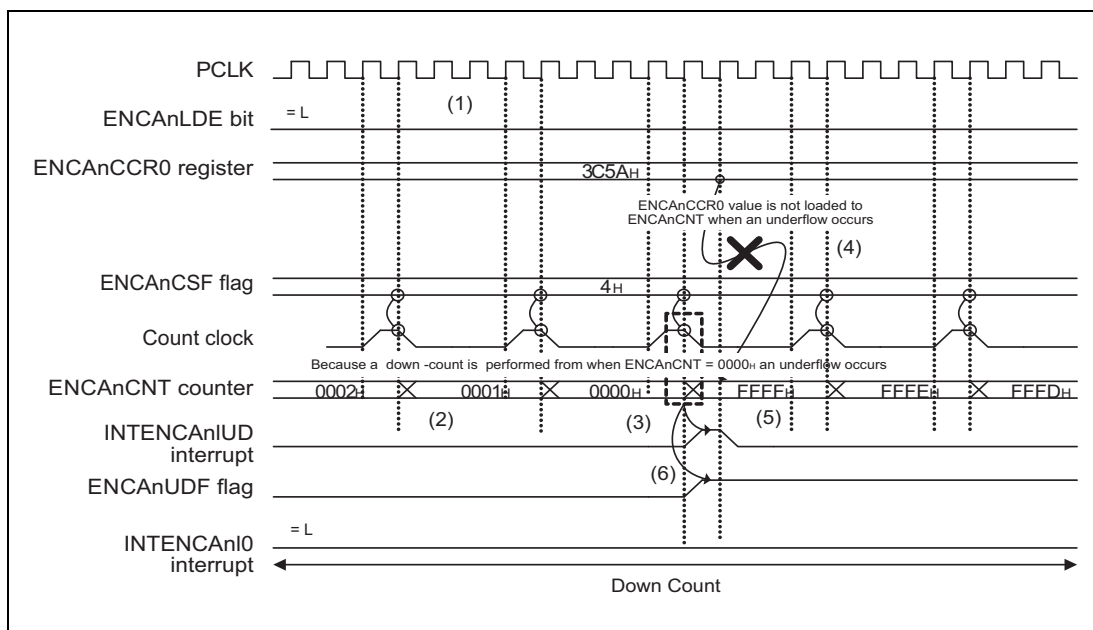
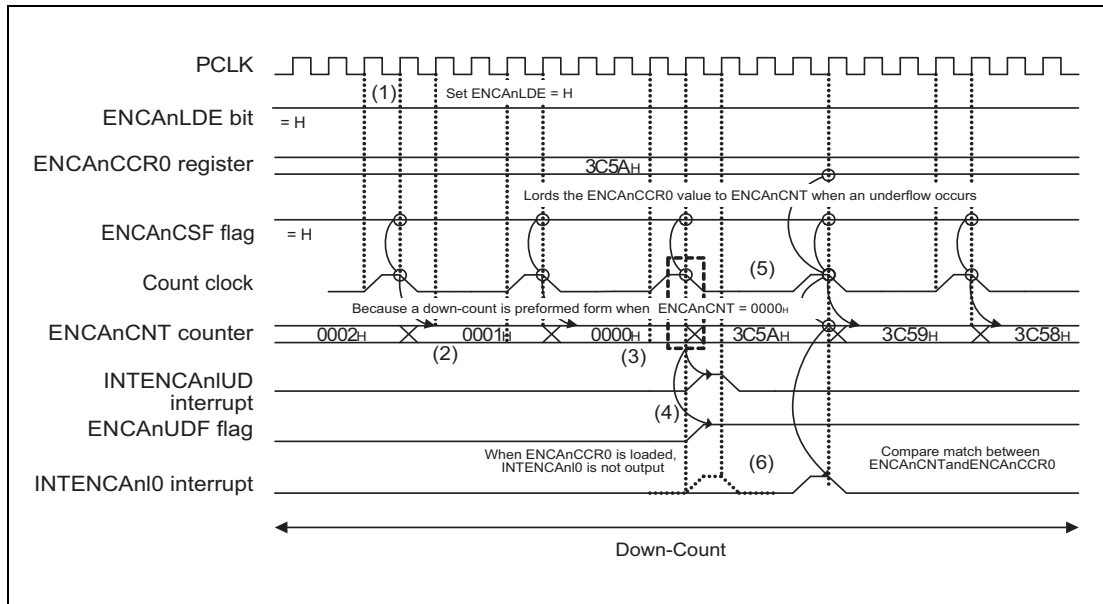


Figure 28.19 ENCA_nLDE Function (when ENCA_nLDE = 0)

- (1) ENCA_nLDE is set to 0 (even if an underflow occurs, the ENCA_nCCR0 value is not loaded).
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.
- (4) Because ENCA_nLDE is set to 0, the setting value of the ENCA_nCCR0 register is not loaded to the counter even if an underflow occurs.
- (5) Operation changes to underflow operation (counter value: 0000_H → FFFF_H).
- (6) An underflow interrupt (INTENCA_nIUD) is output, and the underflow flag (ENCA_nUDF) is set.

(2) <When ENCA_nLDE = 1>Figure 28.20 ENCA_nLDE Function (when ENCA_nLDE = 1)

- (1) ENCA_nLDE is set to 1 (if an underflow occurs, the ENCA_nCCR0 value is loaded to the counter).
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.
- (4) An underflow interrupt is output, and the underflow flag is set.
- (5) Because ENCA_nLDE is set to 1, the setting value of the ENCA_nCCR0 register is loaded to the counter if an underflow occurs. ENCA_nCNT is set to 3C5A_H.
- (6) After the ENCA_nCCR0 value is set to ENCA_nCNT, if the ENCA_nCNT value matches with the ENCA_nCCR0 value on a count clock, a compare match interrupt (INTENCA_nI0) is output.

28.6.10 Conflict between ENCA_nLDE Function (Loading Counter Value) and Rewrite of ENCA_nCCR0 Register

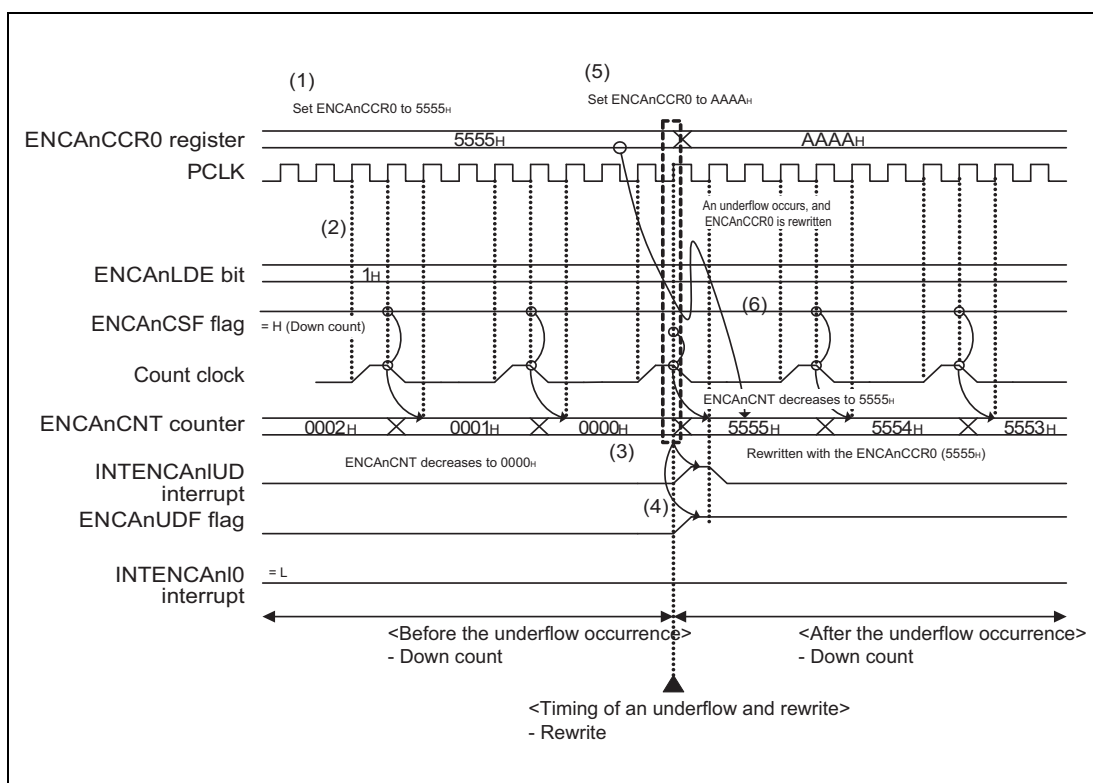


Figure 28.21 Conflict between ENCA_nLDE Function and Rewrite of ENCA_nCCR0 Register

- (1) The ENCA_nCCR0 register is currently set to 5555_H.
- (2) ENCA_nLDE is currently set to 1.
- (3) A down-count is performed (0002_H → 0001_H → 0000_H), and an underflow occurs.
- (4) An underflow interrupt is output, and the underflow flag is set.
- (5) When an underflow occurs, the ENCA_nCCR0 register value is changed from 5555_H to AAAA_H.
- (6) When an underflow occurs, the ENCA_nCCR0 value before the rewrite was performed (5555_H) is set in ENCA_nCNT.

28.6.11 Conflict between ENCA_nLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

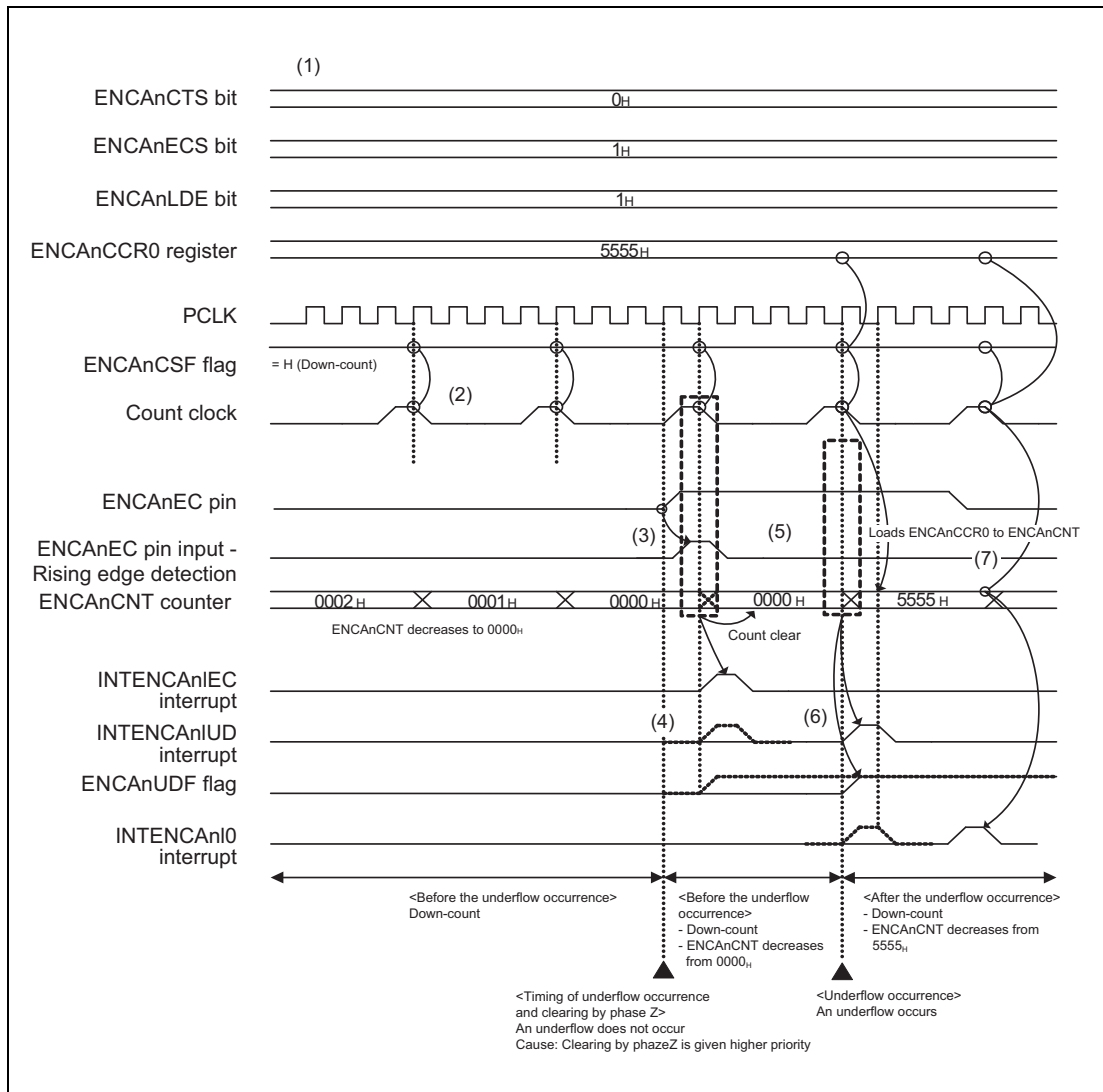


Figure 28.22 Conflict between ENCA_nLDE Function and Clear Operation by Encoder Clear Input

- (1) The values are set as follows: ENCA_nCTS = 0, ENCA_nECS1 and ENCA_nECS0 = 01_B, ENCA_nLDE = 1, and ENCA_nCCR0 = 5555_H.
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When the count value becomes 0000_H, the rising edge of ENCA_nEC is detected, and clear operation by the encoder clear input is performed.
- (4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (INTENCA_nIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (INTENCA_nIUD) is not output, and the underflow flag (ENCA_nUDF) is not set.
- (5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, a down-count is performed and an underflow occurs.
- (6) An underflow interrupt (INTENCA_nIUD) is output, and the underflow flag (ENCA_nUDF) is set.
- (7) Because ENCA_nLDE = 1, if an underflow occurs, the ENCA_nCCR0 value is loaded to ENCA_nCNT.
- (8) After the ENCA_nCCR0 value is set to ENCA_nCNT, a compare match is detected according to the count clock. If the ENCA_nCNT value matches with the ENCA_nCCR0 value, a compare match interrupt (INTENCA_nI0) is output.

28.6.12 Up-count after Conflict between ENCA_nLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

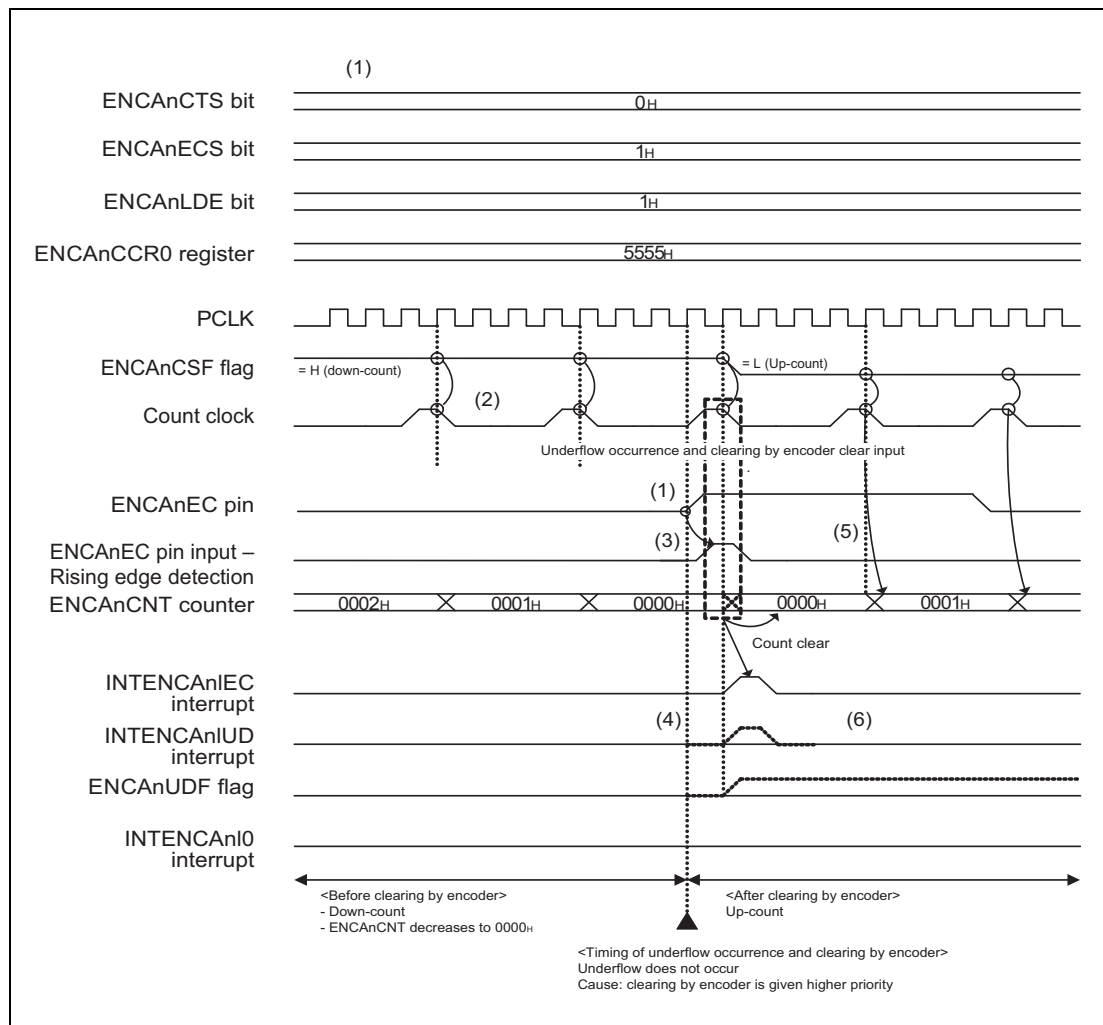


Figure 28.23 Up-count after Conflict between ENCA_nLDE Function and Encoder Clear Input

- (1) The values are set as follows: ENCA_nCTS = 0, ENCA_nECS1 and ENCA_nECS0 = 01_B, ENCA_nLDE = 1, and ENCA_nCCR0 = 5555_H.
- (2) A down-count is performed: 0002_H → 0001_H → 0000_H
- (3) When the count value becomes 0000_H, the rising edge of ENCA_nEC is detected, and clear operation by the encoder clear input is performed.
- (4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (INTENCA_nIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (INTENCA_nUD) is not output, and the underflow flag (ENCA_nUDF) is not set.
- (5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, an up-count is performed.
- (6) An underflow interrupt (INTENCA_nUD) is not output, and the underflow flag (ENCA_nUDF) is not set.

28.6.13 Capture Operation between Count Clocks (ENCA_nCCR1)

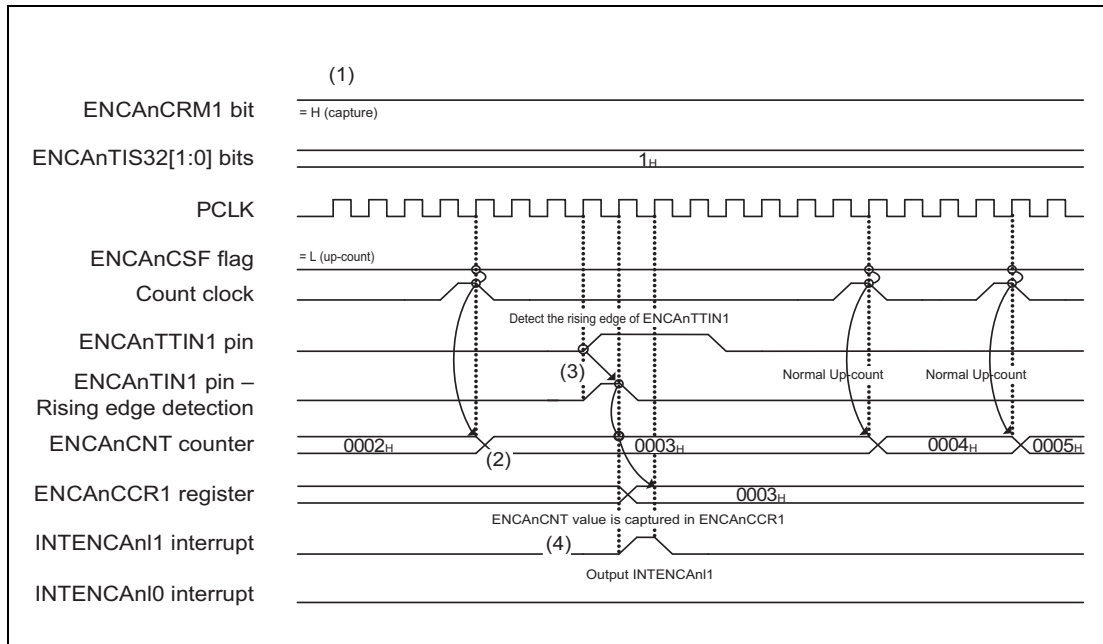


Figure 28.24 Capture Operation between Count Clocks (ENCA_nCCR1)

- (1) The values are set as follows: ENCA_nCRM1 = 1 and ENCA_nTIS32[1:0] = 01_B.
- (2) An up-count is performed.
- (3) The rising edge of the ENCA_nTTIN1 input is detected, and the count value is captured in ENCA_nCCR1.
- (4) An interrupt request signal (INTENCA_n1) corresponding to the capture to the ENCA_nCCR1 register is output.

28.6.14 Capture Operation between Count Clocks (ENCA_nCCR0)

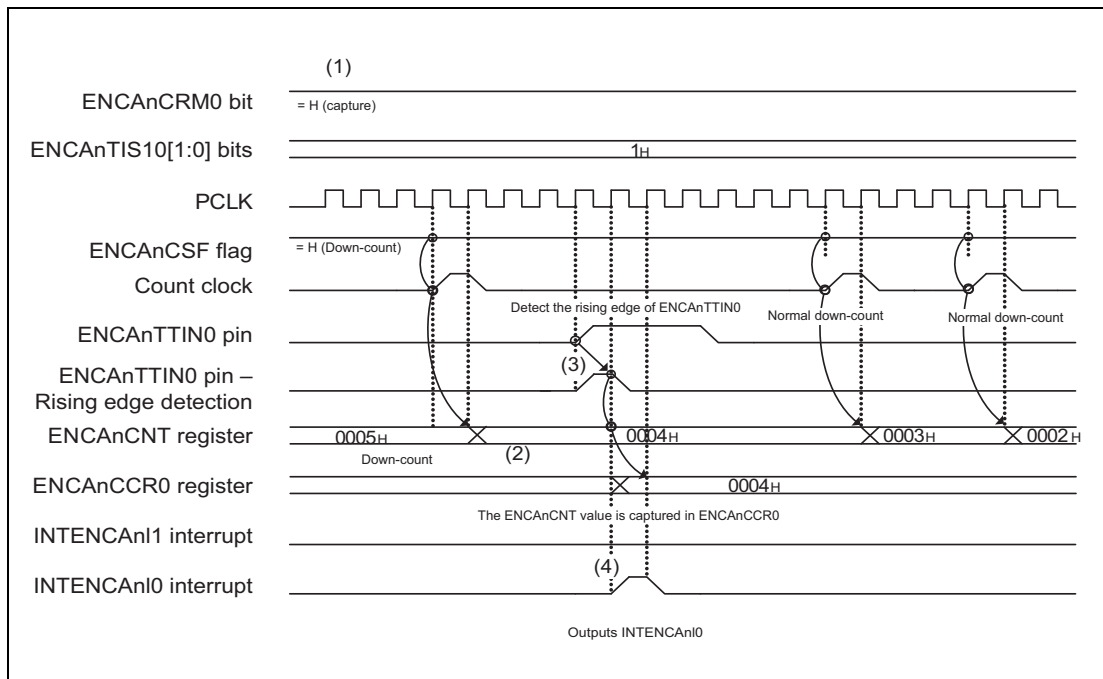


Figure 28.25 Capture Operation between Count Clocks (ENCA_nCCR0)

- (1) The values are set as follows: ENCA_nCRM0 = 1, and ENCA_nTIS10[1:0] = 01_B.
- (2) A down-count is performed.
- (3) The rising edge of the ENCA_nTTIN0 input is detected, and the count value is captured in ENCA_nCCR0.
- (4) An interrupt request signal (INTENCA_n0) corresponding to the capture to the ENCA_nCCR0 register is output.

28.6.15 Encoder operation when compare match clear control is enabled and ENCA_nCTS = 0

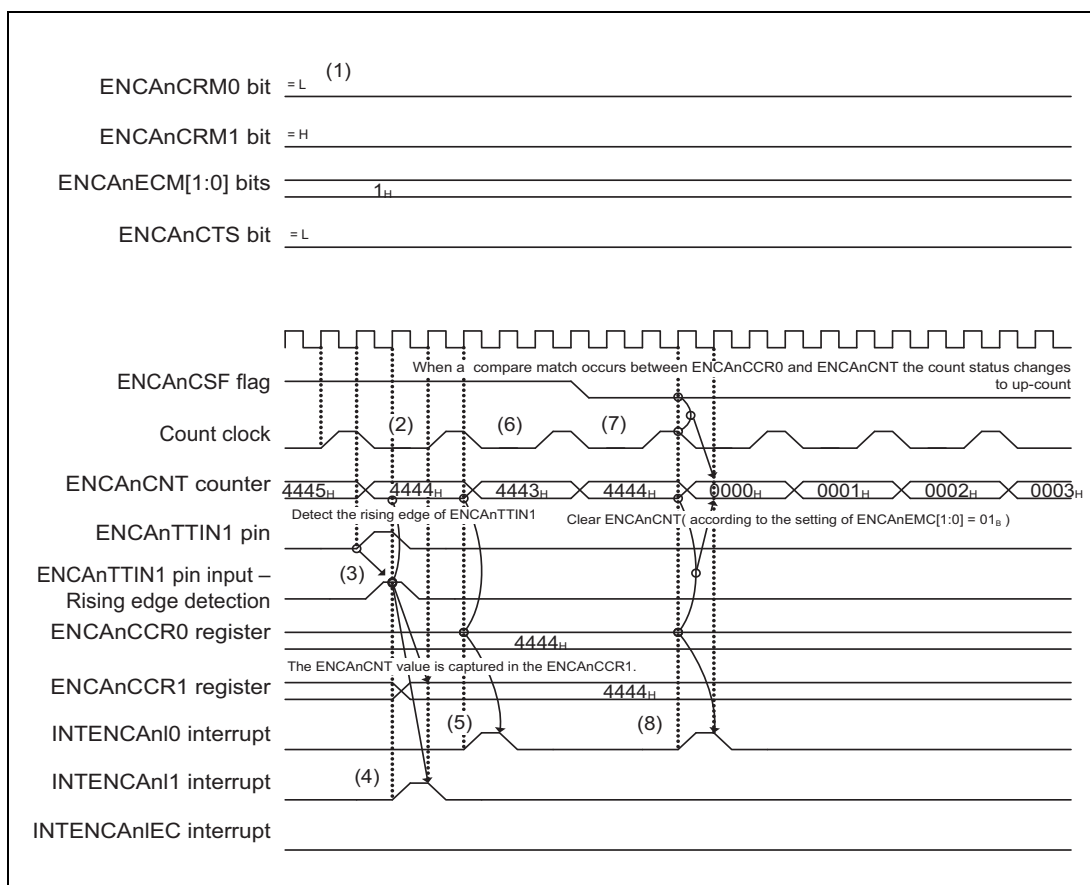


Figure 28.26 Encoder Operation when ENCA_nECM[1:0] = 01_B and ENCA_nCTS = 0

- (1) The values are set as follows: ENCA_nCCR0 = 4444_H, ENCA_nCRM0 = 0, ENCA_nCRM1 = 1, ENCA_nECM[1:0] = 01_B, and ENCA_nCTS = 0.
- (2) A down-count is performed.
- (3) The rising edge of the ENCA_nTTIN1 input is detected, and the ENCA_nCNT value (4444_H) is captured in the ENCA_nCCR1 register.
- (4) An interrupt signal (INTENCA_n1) corresponding to the capture to the ENCA_nCCR1 register is output.
- (5) When a compare match occurs between ENCA_nCNT (counted down from 4445_H to 4444_H) and ENCA_nCCR0 (4444_H), a compare match interrupt (INTENCA_n0) with ENCA_nCCR0 is output.
- (6) The count operation changes to up-count.
- (7) When ENCA_nCNT is counted up from 4443_H to 4444_H, a compare match with ENCA_nCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCA_nECM1 and ENCA_nECM0 (01_B), and the ENCA_nCNT value changes to 0000_H.
- (8) When ENCA_nCNT changes to 4444_H, a compare match interrupt (INTENCA_n0) with ENCA_nCCR0 is output.

28.6.16 Encoder operation when compare match clear control is enabled and ENCA_nCTS = 1

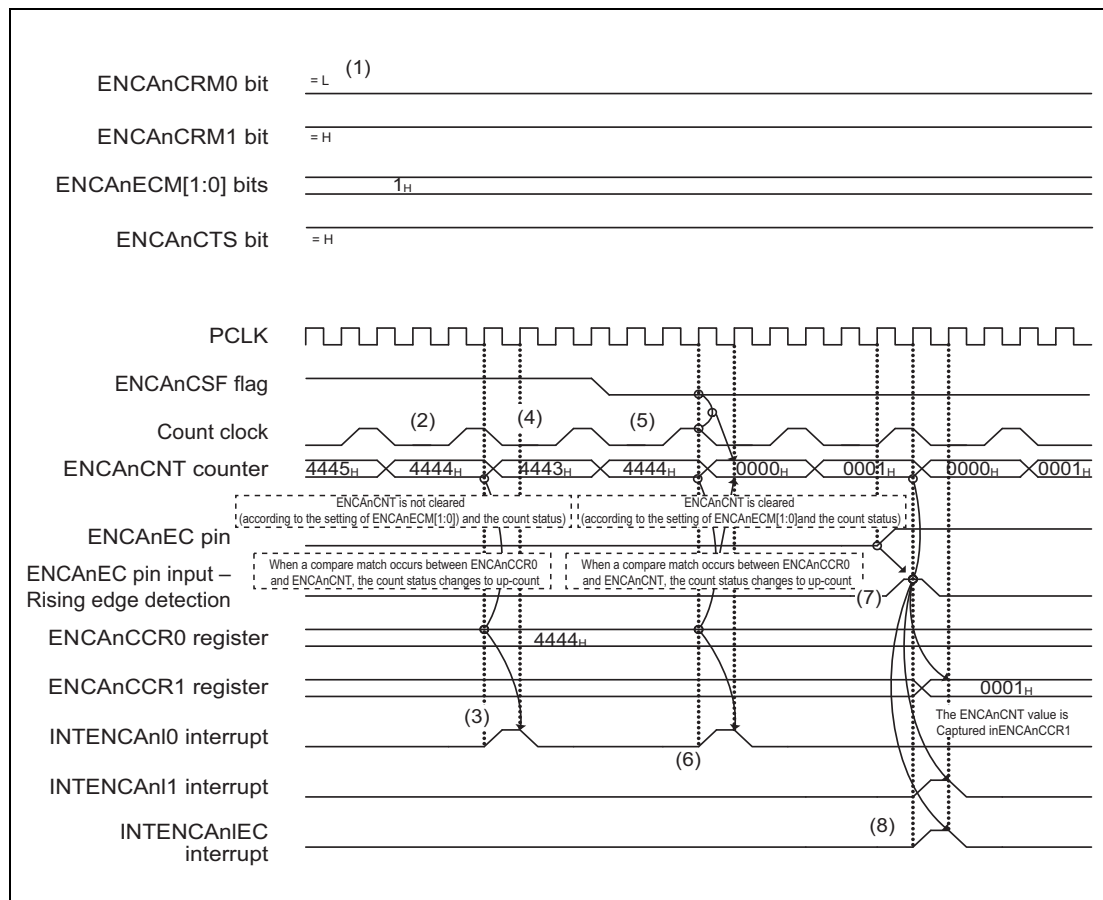


Figure 28.27 Encoder Operation when ENCA_nECM[1:0] = 01_B and ENCA_nCTS = 1

- (1) The values are set as follows: ENCA_nCCR0 = 4444_H, ENCA_nCRM0 = 0, ENCA_nCRM1 = 1, ENCA_nECM[1:0] = 01_B, and ENCA_nCTS = 1.
- (2) A down-count is performed.
- (3) When a compare match occurs between ENCA_nCNT (counted down from 4445_H to 4444_H) and ENCA_nCCR0 (4444_H), an interrupt request signal (INTENCA_nI0) is output.
- (4) The count operation changes to up-count.
- (5) When ENCA_nCNT is counted up from 4443_H to 4444_H, a compare match with ENCA_nCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCA_nECM1 and ENCA_nECM0 (01_B), and the ENCA_nCNT value changes to 0000_H.
- (6) When ENCA_nCNT changes to 4444_H, a compare match interrupt (INTENCA_nI0) with ENCA_nCCR0 is output.
- (7) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H. At this point, the ENCA_nCNT value (0001_H) is captured in ENCA_nCCR1 by detecting the rising edge of the ENCA_nEC signal, and the counter is cleared to 0000_H.
- (8) An interrupt (INTENCA_nI1) corresponding to the capture to the ENCA_nCCR1 register and a clear interrupt (INTENCA_nIEC) by ENCA_nEC are output.

28.6.17 Encoder operation when compare match clear control is disabled

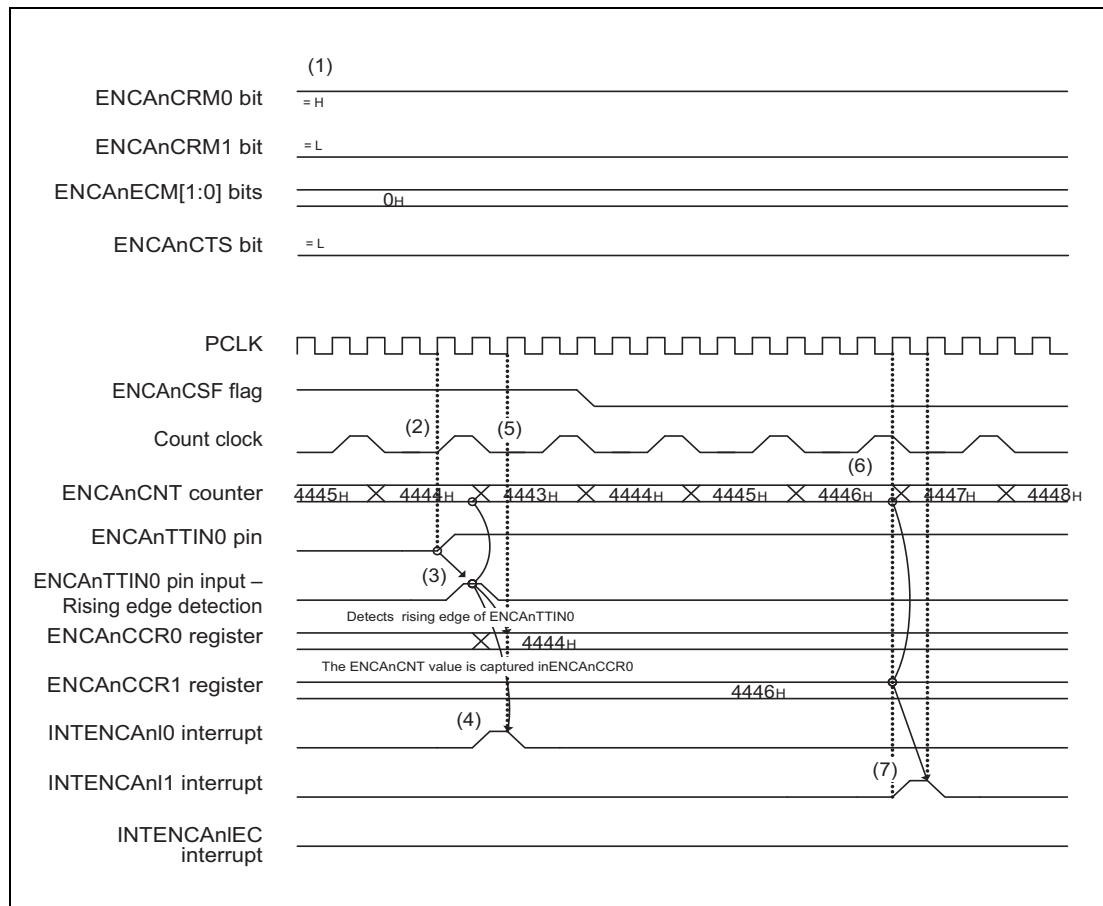


Figure 28.28 Encoder Operation when ENCA nECM[1:0] = 00_B

- (1) The values are set as follows: ENCA nCCR1 = 4446_H, ENCA nCRM0 = 1, ENCA nCRM1 = 0, ENCA nECM[1:0] = 00_B, and ENCA nCTS = 0.
- (2) A down-count is performed.
- (3) When the rising edge of ENCA nTTIN0 is detected, the ENCA nCNT value (4444_H) is captured in ENCA nCCR0.
- (4) An interrupt signal (INTENCA nI0) corresponding to the capture to the ENCA nCCR0 register is output.
- (5) The count operation changes to up-count.
- (6) When ENCA nCNT changes to 4446_H, a compare match with ENCA nCCR1 is detected.
- (7) A compare match interrupt (INTENCA nI1) with ENCA nCCR1 is output.

28.6.18 Capture Operation Performed upon Clearing by ENCA_nEC, ENCA_nE0, and ENCA_nE1 when ENCA_nSCE = 1

28.6.18.1 Accompanying Capture Operation

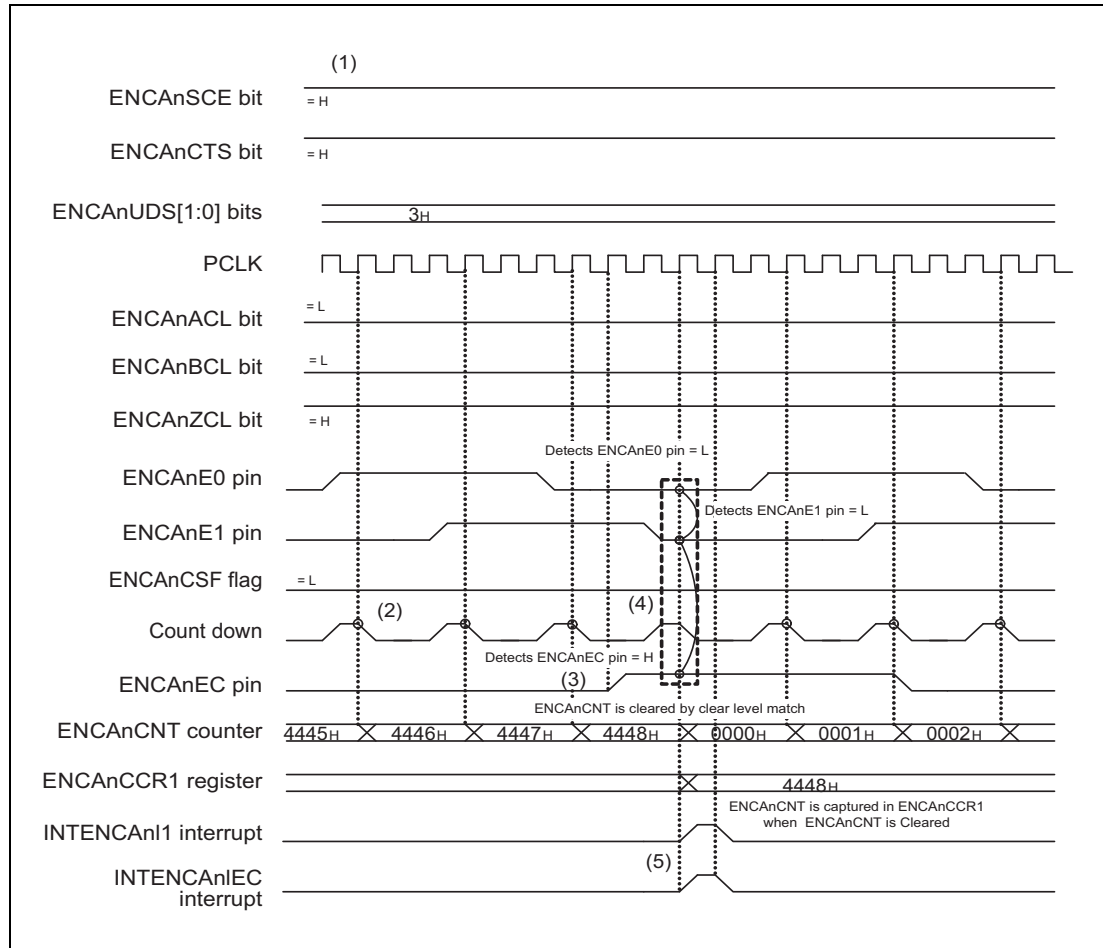


Figure 28.29 Capture Operation Performed upon Clearing by ENCA_nEC, ENCA_nE0, and ENCA_nE1 when ENCA_nSCE = 1

- (1) The values are set as follows: ENCA_nSCE = 1, ENCA_nCTS = 1, ENCA_nUDS[1:0] = 11_B, ENCA_nACL = 0, ENCA_nBCL = 0, and ENCA_nZCL = 1.
- (2) An up-count is performed.
- (3) The count value is not cleared upon the rising edge of ENCA_nEC.
- (4) When ENCA_nE0, ENCA_nE1, and ENCA_nEC reach the set clear level, the count value is cleared. The count value is captured in ENCA_nCCR1 at the time of the clearing.
- (5) At the time of the clearing, an interrupt (INTENCA_n1) corresponding to the capture to the ENCA_nCCR1 register and a clear interrupt (INTENCA_nIEC) by ENCA_nEC are output.

28.6.18.2 When the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Up-count

(When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, and ENCA_nUDS[1:0] = 11_B)

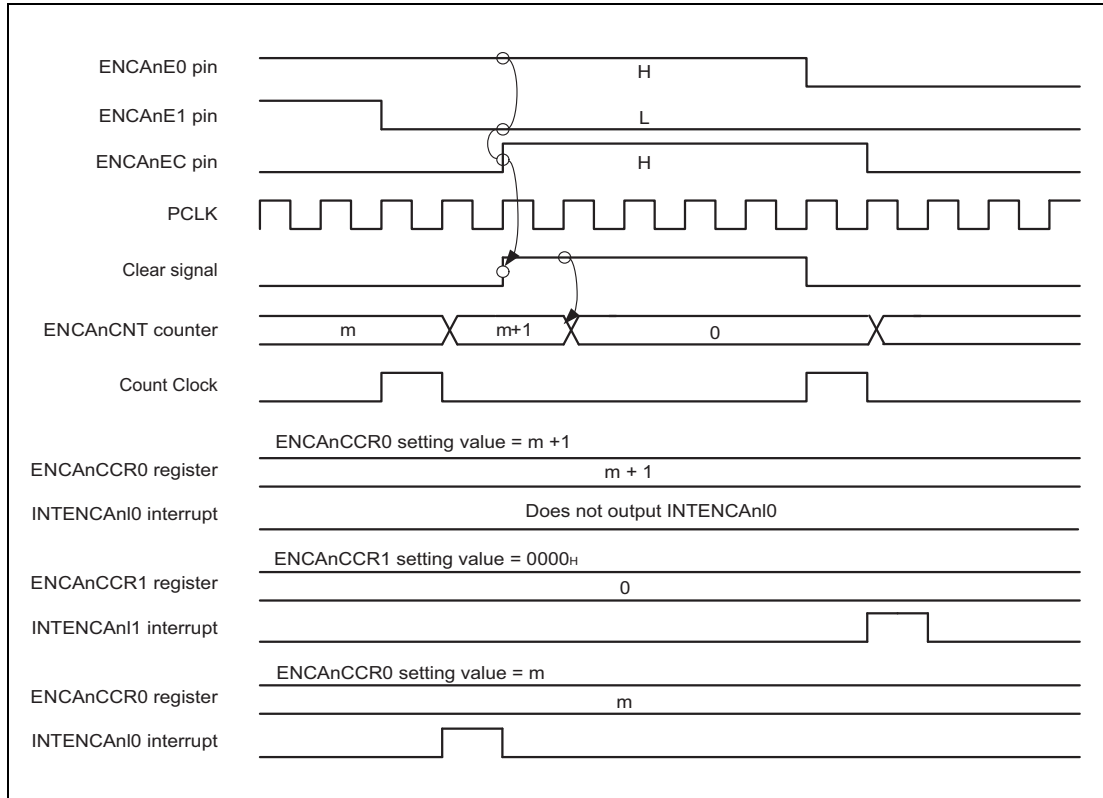


Figure 28.30 Clearing Timing for when the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Up-count

28.6.18.3 When the Timing of the ENCA_nEC Input is the Same as that of the ENCA_nE1 Input during Up-count

(When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, and ENCA_nUDS[1:0] = 11_B)

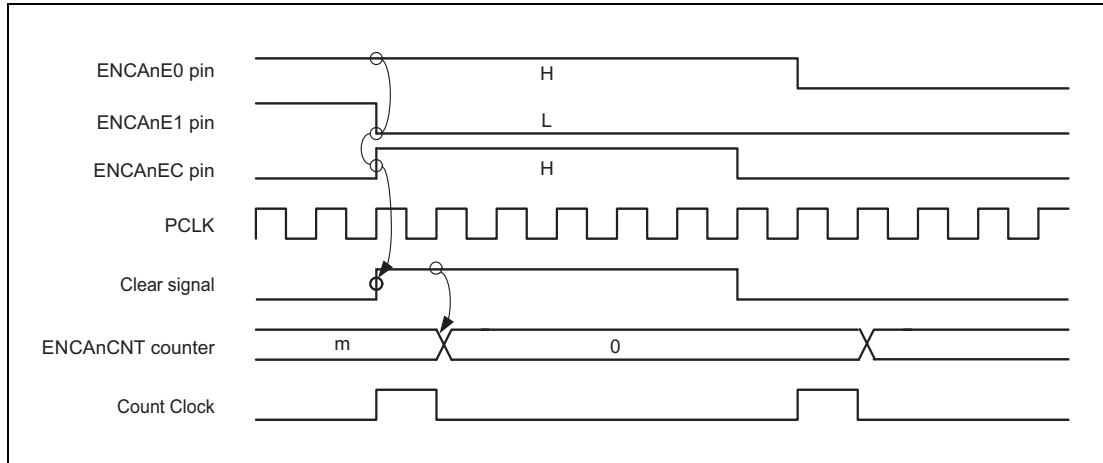


Figure 28.31 Clearing Timing for when the Timing of the ENCA_nEC Input is the Same as that of the ENCA_nE1 Input During Up-count

28.6.18.4 When the Timing of the ENCA_nEC Input is Earlier than that of the ENCA_nE1 Input during Up-count

(When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, and ENCA_nUDS[1:0] = 11_B)

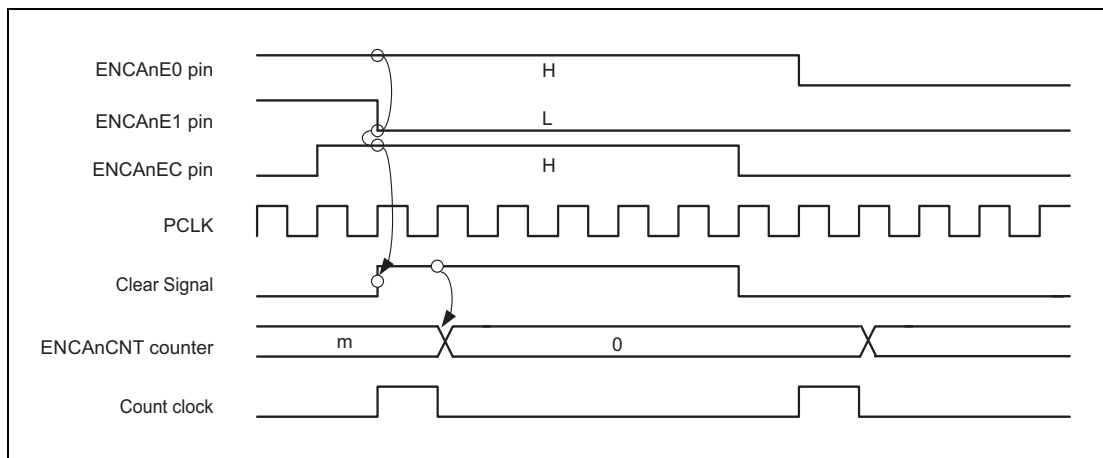


Figure 28.32 Clearing Timing for when the Timing of the ENCA_nEC Input is Earlier than that of the ENCA_nE1 Input during Up-count

28.6.18.5 When the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Down-count

(When ENCA_nACL = 1, ENCA_nBCL = 0, ENCA_nZCL = 1, ENCA_nUDS[1:0] = 11_B)

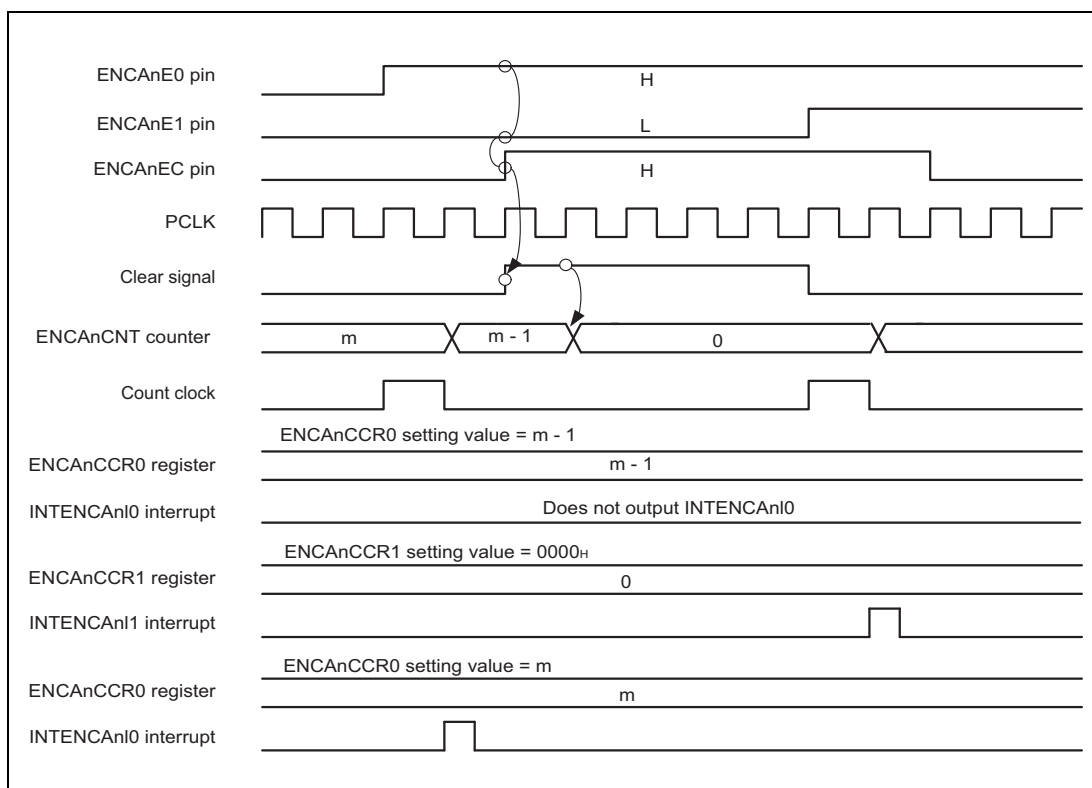


Figure 28.33 Clearing Timing for when the Timing of the ENCA_nEC Input is Later than that of the ENCA_nE1 Input during Down-count

28.6.19 Capture Operation Performed upon Clearing by ENCA_nEC when ENCA_nSCE = 0

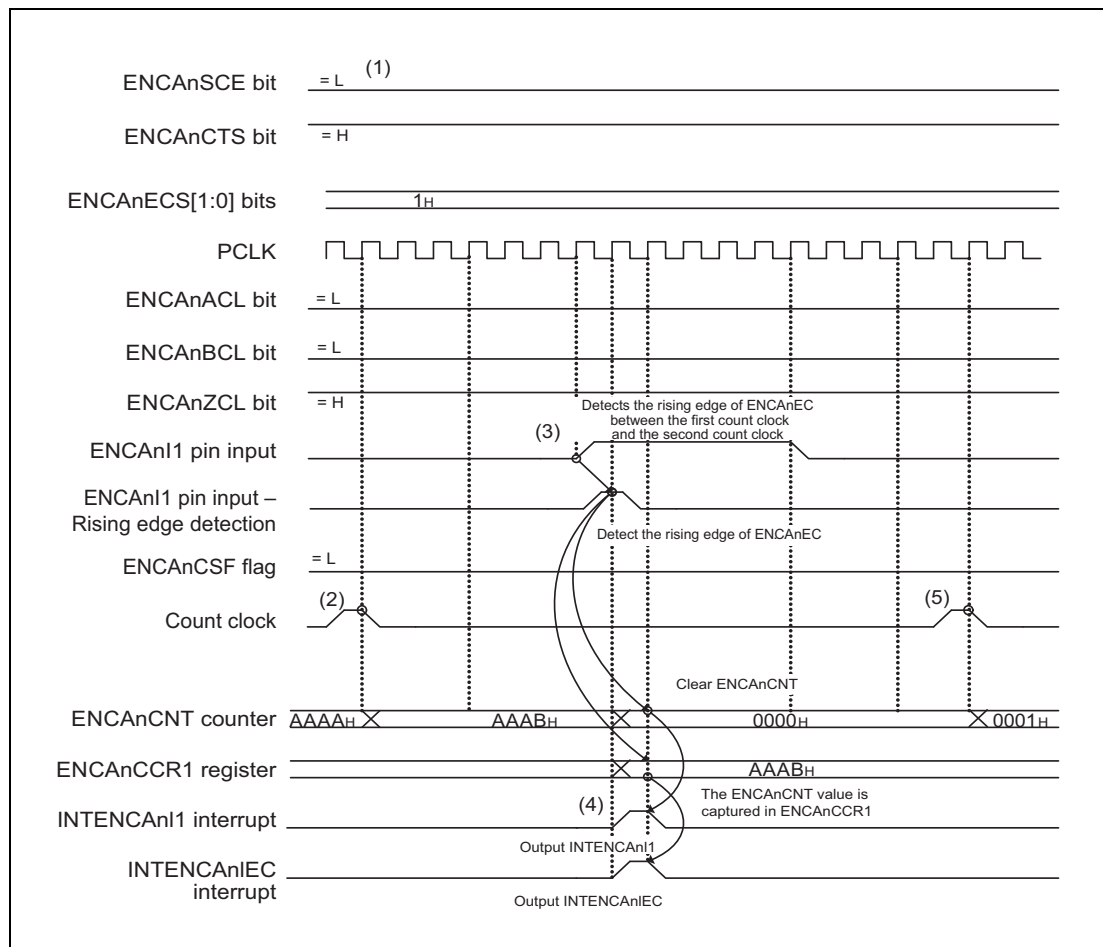


Figure 28.34 Capture Operation Performed upon Clearing by ENCA_nEC when ENCA_nSCE = 0

- (1) The values are set as follows: ENCA_nSCE = 0, ENCA_nCTS = 1, and ENCA_nECS[1:0] = 01_B.
- (2) An up-count is performed.
- (3) The rising edge of the ENCA_nEC input is detected, and the ENCA_nCNT value (AAAB_H) is captured in the ENCA_nCCR1 register. Concurrently, clear operation by ENCA_nEC is performed, and ENCA_nCNT is reset to 0000_H.
- (4) An interrupt (INTENCA_n1) corresponding to the capture to the ENCA_nCCR1 register and an encoder clear interrupt (INTENCA_nIEC) by ENCA_nEC are output.
- (5) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H.

Section 29 Peripheral Interconnection (PIC)

29.1 Features of RH850/P1M-E PIC

Peripheral interconnect (PIC) connects some peripherals with each other in order to achieve enhanced stand-alone functionality. This product includes two PIC units: PIC1A and PIC1B.

29.1.1 Number of Units

This microcontroller has the following number of PIC units.

Table 29.1 Number of Units (PIC1A)

| Products | RH850/P1M-E |
|-----------------|-------------|
| Number of Units | 1 |
| Name | PIC1A |

Table 29.2 Number of Units (PIC2B)

| Products | RH850/P1M-E |
|-----------------|-------------|
| Number of Units | 1 |
| Name | PIC2B |

Table 29.3 Index

| Index | Meaning |
|-------|--|
| n | The individual unit of each timer and A/D converter is identified by the index "n". |
| m | The individual channel of each timer and A/D converter is identified by the index "m". |
| x | The scan group number of A/D converter is indicated by the index "x". |
| i | The variable used for description is indicated by the index "i". |

29.1.2 Register Base Address

PIC base addresses are listed in the following table.

PIC register addresses are given as offsets from the base addresses.

Table 29.4 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <PIC1A_base> | FFDD 0000 _H |
| <PIC2B_base> | FFDD 1000 _H |

29.1.3 Clock Supply

Clock supplies by and to PIC are listed in the following table.

Table 29.5 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|---------------------------------------|
| PIC1A | PCLK | High-speed peripheral clock (CLK_HSB) |
| PIC2B | PCLK | High-speed peripheral clock (CLK_HSB) |

29.1.4 Reset Sources

Please refer to **Section 8, Reset Controller**.

29.1.5 Input/Output Signals

Table 29.6 PIC Input/Output Signals (1/2)

| Signal Name | I/O | Function | Connected to |
|--------------------------|-----|---|--------------|
| ENCA0TSST | O | ENCA0 simultaneous start trigger signal | ENCA0 |
| ENCAT0AIN | O | ENCA0 encoder input signal (A phase) | |
| ENCAT0BIN | O | ENCA0 encoder input signal (B phase) | |
| ENCAT0ZIN | O | ENCA0 encoder input signal (Z phase) | |
| INTENCA0I1 | O | ENCA0 capture trigger | |
| INTENCA0IEC | I | Clear interrupt signal by the ENCA0 encoder input (Z phase) | |
| ENCAT0EQ0 | I | ENCA0 match detection signal 0 | |
| ENCAT0EQ1 | I | ENCA0 match detection signal 1 | |
| ENCA1TSST | O | ENCA1 simultaneous start trigger signal | ENCA1 |
| ENCAT1AIN | O | ENCA1 encoder input signal (A phase) | |
| ENCAT1BIN | O | ENCA1 encoder input signal (B phase) | |
| ENCAT1ZIN | O | ENCA1 encoder input signal (Z phase) | |
| INTENCA1I1 | O | ENCA1 capture trigger | |
| INTENCA1IEC | I | Clear interrupt signal by the ENCA1 encoder input (Z phase) | |
| ENCAT1EQ0 | I | ENCA1 match detection signal 0 | |
| ENCAT1EQ1 | I | ENCA1 match detection signal 1 | |
| TOP0TAPATHASIN | O | Hi-Z asynchronous signal | TAPA0 |
| TOP1TAPATHASIN | O | Hi-Z asynchronous signal | TAPA1 |
| TOP2TAPATHASIN | O | Hi-Z asynchronous signal | TAPA2 |
| TOP3TAPATHASIN | O | Hi-Z asynchronous signal | TAPA3 |
| TAUD0TSSTm (m = 0 to 15) | O | TAUD0 simultaneous start trigger signal | TAUD0 |
| TAUD0TOUTm (m = 0 to 15) | I | TAUD0 channel output signal | |
| TAUD0TINm (m = 0 to 15) | O | TAUD0 channel input signal | |
| INTTAUD0Im (m = 0 to 15) | I | TAUD0 interrupt signal | |
| TAUD1TSSTm (m = 0 to 15) | O | TAUD1 simultaneous start trigger signal | TAUD1 |
| TAUD1TOUTm (m = 0 to 15) | I | TAUD1 channel output signal | |
| TAUD1TINm (m = 0 to 15) | O | TAUD1 channel input signal | |
| INTTAUD1Im (m = 0 to 15) | I | TAUD1 interrupt signal | |

Table 29.6 PIC Input/Output Signals (2/2)

| Signal Name | I/O | Function | Connected to |
|--------------------------------------|-----|---|--------------|
| TAUJ0TSST _m (m = 0 to 3) | O | TAUJ0 simultaneous start trigger signal | TAUJ0 |
| TAUJ0TIN _m (m = 0 to 3) | O | TAUJ0 channel input signal | |
| TAUJ1TSST _m (m = 0 to 3) | O | TAUJ1 simultaneous start trigger signal | TAUJ1 |
| TPBA0TSST | O | TPBA0 simultaneous start trigger signal | TPBA0 |
| TPBA1TSST | O | TPBA1 simultaneous start trigger signal | TPBA1 |
| TSG30TSST | O | TSG30 simultaneous start trigger signal | TSG30 |
| TSG30TSTOPC0 | O | Timer output pattern control 0 signal | |
| TSG30TSTOPC1 | O | Timer output pattern control 1 signal | |
| TSG30TSTPTE | I | TSTAP[0:2] edge detection signal | |
| TSG30TS2PEC | I | Two-phase encoder count signal | |
| TSG30TST2PUD | I | Two-phase encoder up/down signal | |
| TSG30TSTAPT0 | O | Hall sensor input signal 0 | |
| TSG30TSTAPT1 | O | Hall sensor input signal 1 | |
| TSG30TSTAPT2 | O | Hall sensor input signal 2 | |
| TSG30TO1 | I | PWM output 1 | |
| TSG30TO2 | I | PWM output 2 | |
| TSG30TO3 | I | PWM output 3 | |
| TSG30TO4 | I | PWM output 4 | |
| TSG30TO5 | I | PWM output 5 | |
| TSG30TO6 | I | PWM output 6 | |
| TSG31TSST | O | TSG31 simultaneous start trigger signal | TSG31 |
| TSG31TSTOPC0 | O | Timer output pattern control 0 signal | |
| TSG31TSTOPC1 | O | Timer output pattern control 1 signal | |
| TSG31TSTPTE | I | TSTAP[0:2] edge detection signal | |
| TSG31TS2PEC | I | Two-phase encoder count signal | |
| TSG31TST2PUD | I | Two-phase encoder up/down signal | |
| TSG31TSTAPT0 | O | Hall sensor input signal 0 | |
| TSG31TSTAPT1 | O | Hall sensor input signal 1 | |
| TSG31TSTAPT2 | O | Hall sensor input signal 2 | |
| TSG31TO1 | I | PWM output 1 | |
| TSG31TO2 | I | PWM output 2 | |
| TSG31TO3 | I | PWM output 3 | |
| TSG31TO4 | I | PWM output 4 | |
| TSG31TO5 | I | PWM output 5 | |
| TSG31TO6 | I | PWM output 6 | |
| ADCGTTOUT0 _m (m = 0 to 4) | I | A/D converter 0 hardware trigger | PIC2B |
| ADCGTTOUT1 _m (m = 0 to 4) | I | A/D converter 1 hardware trigger | |
| OSTM0TSST | O | OSTM0 simultaneous start trigger signal | OSTM0 |
| OSTM1TSST | O | OSTM1 simultaneous start trigger signal | OSTM1 |
| INTTSG30IER | I | Hi-Z control signal by TSG30 error interrupt | TSG30 |
| INTTSG31IER | I | Hi-Z control signal by TSG31 error interrupt | TSG31 |
| ERROROUT | I | ERROR output signal | ECM |
| INTADCG0ERR | I | ADCG0 error interrupt signal for Hi-Z control | ADCG0 |
| INTADCG1ERR | I | ADCG1 error interrupt signal for Hi-Z control | ADCG1 |

29.1.6 External Input/Output Signals

External input/output signals of PIC are listed in the following table.

Table 29.7 PIC1A External Input/Output Signals (1/2)

| Unit Signal Name | Description | Alternative Port Pin Signal Name |
|----------------------------|-------------------------------------|----------------------------------|
| ENCA0I1 | ENCA0 capture trigger input 1 | ENCA0TIN1 |
| ENCA1I1 | ENCA1 capture trigger input 1 | ENCA1TIN1 |
| ENCA0E0 | ENCA0 encoder input (count pulse 0) | ENCA0E0 |
| ENCA1E0 | ENCA1 encoder input (count pulse 0) | ENCA1E0 |
| ENCA0E1 | ENCA0 encoder input (count pulse 1) | ENCA0E1 |
| ENCA1E1 | ENCA1 encoder input (count pulse 1) | ENCA1E1 |
| ENCA0EC | ENCA0 encoder input (clear pulse) | ENCA0EC |
| ENCA1EC | ENCA1 encoder input (clear pulse) | ENCA1EC |
| TAUD0TINm (m = 0 to 15) | TAUD0 channel input m (m = 0 to 15) | TAUD0Im (m = 0 to 15) |
| TAUD1TINm (m = 0 to 15) | TAUD1 channel input m (m = 0 to 15) | TAUD1Im (m = 0 to 15) |
| TAUD2TINm (m = 0 to 15) | TAUD2 channel input m (m = 0 to 15) | TAUD2Im (m = 0 to 15) |
| ESOn | Hi-Z control | TAPAnESO |
| TOP0U | Motor control output U phase | TAUD0O10 |
| TOP0UB | Motor control output UB phase | TAUD0O11 |
| TOP0V | Motor control output V phase | TAUD0O12 |
| TOP0VB | Motor control output VB phase | TAUD0O13 |
| TOP0W | Motor control output W phase | TAUD0O14 |
| TOP0WB | Motor control output WB phase | TAUD0O15 |
| TOP1U | Motor control output U phase | TAUD1O10 |
| TOP1UB | Motor control output UB phase | TAUD1O11 |
| TOP1V | Motor control output V phase | TAUD1O12 |
| TOP1VB | Motor control output VB phase | TAUD1O13 |
| TOP1W | Motor control output W phase | TAUD1O14 |
| TOP1WB | Motor control output WB phase | TAUD1O15 |
| TOP2U | Motor control output U phase | TSG3001 |
| TOP2UB | Motor control output UB phase | TSG3002 |
| TOP2V | Motor control output V phase | TSG3003 |
| TOP2VB | Motor control output VB phase | TSG3004 |
| TOP2W | Motor control output W phase | TSG3005 |
| TOP2WB | Motor control output WB phase | TSG3006 |
| TOP3U | Motor control output U phase | TSG3101 |
| TOP3UB | Motor control output UB phase | TSG3102 |
| TOP3V | Motor control output V phase | TSG3103 |
| TOP3VB | Motor control output VB phase | TSG3104 |
| TOP3W | Motor control output W phase | TSG3105 |
| TOP3WB | Motor control output WB phase | TSG3106 |

Table 29.7 PIC1A External Input/Output Signals (2/2)

| Unit Signal Name | Description | Alternative Port Pin Signal Name |
|---------------------------|-------------|----------------------------------|
| TAUJ0TINm (m = 0 to 3) | TAUJ input | TAUJ0Im (m = 0 to 3) |
| TAUJ1TINm (m = 0 to 3) | TAUJ input | TAUJ1Im (m = 0 to 3) |
| TAUJ2TINm (m = 0 to 3) | TAUJ input | TAUJ2Im (m = 0 to 3) |

Table 29.8 PIC2B External Input/Output Signals

| Unit Signal Name | Description | Alternative Port Pin Signal Name |
|------------------|---------------|----------------------------------|
| ADTRG0Z | ADCG0 trigger | ADCGTRG0 |
| ADTRG1Z | ADCG1 trigger | ADCGTRG1 |

29.2 Peripheral Interconnection — 1 (PIC1A)

29.2.1 Overview

29.2.1.1 Functional Overview

The peripheral interconnection-1 (PIC1A) realizes various functions by synchronous operation using multiple timers and by connecting the timer internal I/O signals between the timers.

CAUTION

The signal names used in the following descriptions are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

INTm → TAUDnTINTm

TINm → TAUDnTTINm

TOUTm → TAUDnTTOUTm

CDRm → TAUDnCDRm

CNTm → TAUDnCNTm

The PIC1A has the following functions.

- Simultaneous start trigger function
- PWM output function with dead time
- High accuracy triangle wave PWM output function with dead time
- Delay pulse output function with dead time
- Trigger pulse width measurement function
- Encoder capture trigger select function
- Two-phase encoder control function (control method 1)
- Two-phase encoder control function (control method 2)
- Three-phase pulse input control function
- Three-phase encoder control function
- TAUD input select function
- Hi-Z control function
- Timer output monitor function (PWM-Diag)
- Timer input monitor function
- TSG3 Synchronous Clear Function

29.2.2 Registers

29.2.2.1 List of Registers

The registers are listed in the following table.

The bits can be accessed only in 32-bit units. Access in 16-bits or 8-bits is operated as 32-bit access.

Table 29.9 Registers

| Register Function | Symbol | Address |
|---|-----------------|--------------------------------|
| Simultaneous start trigger control register | PIC1ASST | <PIC1A_base> + 04 _H |
| Simultaneous start control register 0 | PIC1ASSER0 | <PIC1A_base> + 10 _H |
| Simultaneous start control register 1 | PIC1ASSER1 | <PIC1A_base> + 14 _H |
| Simultaneous start control register 2 | PIC1ASSER2 | <PIC1A_base> + 18 _H |
| Simultaneous start control register 3 | PIC1ASSER3 | <PIC1A_base> + 1C _H |
| RS flip-flop circuit initialization register 00 | PIC1AINI00 | <PIC1A_base> + 20 _H |
| DT initialization register 01 | PIC1AINI01 | <PIC1A_base> + 24 _H |
| RS flip-flop circuit initialization register 10 | PIC1AINI10 | <PIC1A_base> + 2C _H |
| DT initialization register 11 | PIC1AINI11 | <PIC1A_base> + 30 _H |
| Hall sensor input select register | PIC1ATSGHALLSEL | <PIC1A_base> + 74 _H |
| TAUD0 input select register | PIC1ATAUD0SEL | <PIC1A_base> + 78 _H |
| TAUD1 input select register | PIC1ATAUD1SEL | <PIC1A_base> + 7C _H |
| Hi-Z control register 0 | PIC1AHIZCEN0 | <PIC1A_base> + 80 _H |
| Hi-Z control register 1 | PIC1AHIZCEN1 | <PIC1A_base> + 84 _H |
| Hi-Z control register 2 | PIC1AHIZCEN2 | <PIC1A_base> + 88 _H |
| Hi-Z control register 3 | PIC1AHIZCEN3 | <PIC1A_base> + 8C _H |
| ENCATIN1 input select register 400 | PIC1AENCSEL400 | <PIC1A_base> + B8 _H |
| ENCATIN1 input select register 410 | PIC1AENCSEL410 | <PIC1A_base> + BC _H |
| Timer input/output control register 200 | PIC1AREG200 | <PIC1A_base> + C0 _H |
| Timer input/output control register 201 | PIC1AREG201 | <PIC1A_base> + C4 _H |
| Timer input/output control register 202 | PIC1AREG202 | <PIC1A_base> + C8 _H |
| Timer input/output control register 203 | PIC1AREG203 | <PIC1A_base> + CC _H |
| Timer input/output control register 210 | PIC1AREG210 | <PIC1A_base> + D4 _H |
| Timer input/output control register 211 | PIC1AREG211 | <PIC1A_base> + D8 _H |
| Timer input/output control register 212 | PIC1AREG212 | <PIC1A_base> + DC _H |
| Timer input/output control register 213 | PIC1AREG213 | <PIC1A_base> + E0 _H |
| Timer input/output control register 30 | PIC1AREG30 | <PIC1A_base> + E8 _H |
| Timer input/output control register 31 | PIC1AREG31 | <PIC1A_base> + EC _H |
| Timer input/output control register 50 | PIC1AREG50 | <PIC1A_base> + F8 _H |
| Timer input/output control register 51 | PIC1AREG51 | <PIC1A_base> + FC _H |
| Synchronous clear enable register | SELBSSER | FFDD 2000 _H |
| Port output monitor select register | POMONSEL | FFDD 7400 _H |
| Port input monitor select register | PIMONSEL | FFDD 7000 _H |

Combinations of registers used for each function are listed in the following table.

Table 29.10 Registers Used by Each Function

| Section Number | Function Name | PIC1ASST | | | | PIC1ASSER | | | | PIC1AINI | | | | PIC1ATSGHALLSEL | PIC1ATAUD0SEL | PIC1ATAUD1SEL | PIC1AHIZCE | | | | PIC1AENCSEL4 | | | | PIC1AREG | | | | | | | | | | | | POMONSEL | PIMONSEL | SELSSER | |
|----------------|--|----------|---|---|---|-----------|---|---|---|----------|----|----|----|-----------------|---------------|---------------|------------|---|---|---|--------------|----|-----|-----|----------|-----|-----|-----|-----|-----|----|----|----|----|---|---|----------|----------|---------|---|
| | | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 00 | 01 | 10 | 11 | — | — | — | 0 | 1 | 2 | 3 | 00 | 10 | 200 | 201 | 202 | 203 | 210 | 211 | 212 | 213 | 30 | 31 | 50 | 51 | — | — | — | | | |
| 29.2.3.1 | Simultaneous Start Trigger Function | √ | √ | √ | √ | — | — | — | — | — | — | — | — | — | √ | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 29.2.3.2 | PWM Output Function with Dead Time | — | — | — | — | — | — | — | — | — | — | √ | — | — | √ | √ | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 29.2.3.3 | High Accuracy Triangle Wave PWM Output Function with Dead Time | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 29.2.3.4 | Delay Pulse Output Function with Dead Time | — | — | — | — | — | — | — | — | — | — | — | — | — | √ | √ | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 29.2.3.5 | Trigger Pulse Interval Measurement Function | — | — | — | — | — | — | — | — | — | — | — | √ | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.6 | Encoder Capture Trigger Select Function | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.7 | Two-Phase Encoder Control Function (Control Method 1) | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.8 | Two-Phase Encoder Control Function (Control Method 2) | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.9 | Three-Phase Pulse Input Control Function | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.10 | Three-Phase Encoder Control Function | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.11 | TAUD Input Select Function | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.12 | Hi-Z Control Function | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.13 | Timer Output Monitor Function (PWM-Diag) | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.14 | Timer Input Monitor Function | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| 29.2.3.15 | TSG3 Synchronous clear function | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |

29.2.2.2 PIC1ASST — Simultaneous Start Trigger Control Register

The PIC1ASST register is an 8-bit register that selects the simultaneous start trigger.

Access: This register can be read/written in 8-bit units.

Address: FFDD 0004_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|------------------|
| | — | — | — | — | — | — | — | PIC1ASYNCTR G |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 29.11 PIC1ASST Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PIC1ASYNCTR G | Generates a start trigger for the timer for which simultaneous start is enabled. 0: Disabled 1: Simultaneous start trigger (pulse with a width of 1PCLK is output). |

Note: PIC1ASYNCTR_G always reads 0 when read.

29.2.2.3 PIC1ASSER0 — Simultaneous Start Control Register 0

The PIC1ASSER0 register enables a start trigger for each channel of TAUD0.

Access: This register can be read/written in 16-bit units.

Address: FFDD 0010_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | PIC1AS SER015 | PIC1AS SER014 | PIC1AS SER013 | PIC1AS SER012 | PIC1AS SER011 | PIC1AS SER010 | PIC1AS SER009 | PIC1AS SER008 | PIC1AS SER007 | PIC1AS SER006 | PIC1AS SER005 | PIC1AS SER004 | PIC1AS SER003 | PIC1AS SER002 | PIC1AS SER001 | PIC1AS SER000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.12 PIC1ASSER0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 15 to 0 | PIC1ASSER0m | Enables or disables a simultaneous start trigger for CHm of TAUD0. 0: Disabled 1: Enabled |

29.2.2.4 PIC1ASSER1 — Simultaneous Start Control Register 1

The PIC1ASSER1 register enables a start trigger for each channel of TAUD1.

Access: This register can be read/written in 16-bit units.

Address: FFDD 0014_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | PIC1AS SER115 | PIC1AS SER114 | PIC1AS SER113 | PIC1AS SER112 | PIC1AS SER111 | PIC1AS SER110 | PIC1AS SER109 | PIC1AS SER108 | PIC1AS SER107 | PIC1AS SER106 | PIC1AS SER105 | PIC1AS SER104 | PIC1AS SER103 | PIC1AS SER102 | PIC1AS SER101 | PIC1AS SER100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.13 PIC1ASSER1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 15 to 0 | PIC1ASSER1m | Enables or disables a simultaneous start trigger for CHm of TAUD1. 0: Disabled 1: Enabled |

29.2.2.5 PIC1ASSER2 — Simultaneous Start Control Register 2

The PIC1ASSER2 register enables a start trigger of TAUJ_n, TSG3_n, TPBA_n, and ENCA_n.

Access: This register can be read/written in 16-bit units.

Address: FFDD 0018_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | — | — | PIC1ASSER213 | PIC1ASSER212 | PIC1ASSER211 | PIC1ASSER210 | PIC1ASSER209 | PIC1ASSER208 | PIC1ASSER207 | PIC1ASSER206 | PIC1ASSER205 | PIC1ASSER204 | PIC1ASSER203 | PIC1ASSER202 | PIC1ASSER201 | PIC1ASSER200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.14 PIC1ASSER2 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 15, 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 13 | PIC1ASSER213 | Enables or disables a simultaneous start trigger for the ENCA1. 0: Disabled 1: Enabled |
| 12 | PIC1ASSER212 | Enables or disables a simultaneous start trigger for the ENCA0. 0: Disabled 1: Enabled |
| 11 | PIC1ASSER211 | Enables or disables a simultaneous start trigger for the TPBA1. 0: Disabled 1: Enabled |
| 10 | PIC1ASSER210 | Enables or disables a simultaneous start trigger for the TPBA0. 0: Disabled 1: Enabled |
| 9 | PIC1ASSER209 | Enables or disables a simultaneous start trigger for the TSG31. 0: Disabled 1: Enabled |
| 8 | PIC1ASSER208 | Enables or disables a simultaneous start trigger for the TSG30. 0: Disabled 1: Enabled |
| 7 | PIC1ASSER207 | Enables or disables a simultaneous start trigger for CH03 of TAUJ1. 0: Disabled 1: Enabled |
| 6 | PIC1ASSER206 | Enables or disables a simultaneous start trigger for CH02 of TAUJ1. 0: Disabled 1: Enabled |
| 5 | PIC1ASSER205 | Enables or disables a simultaneous start trigger for CH01 of TAUJ1. 0: Disabled 1: Enabled |
| 4 | PIC1ASSER204 | Enables or disables a simultaneous start trigger for CH00 of TAUJ1. 0: Disabled 1: Enabled |
| 3 | PIC1ASSER203 | Enables or disables a simultaneous start trigger for CH03 of TAUJ0. 0: Disabled 1: Enabled |
| 2 | PIC1ASSER202 | Enables or disables a simultaneous start trigger for CH02 of TAUJ0. 0: Disabled 1: Enabled |
| 1 | PIC1ASSER201 | Enables or disables a simultaneous start trigger for CH01 of TAUJ0. 0: Disabled 1: Enabled |

Table 29.14 PIC1ASSER2 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 0 | PIC1ASSER200 | Enables or disables a simultaneous start trigger for CH00 of TAUJ0. 0: Disabled 1: Enabled |

29.2.2.6 PIC1ASSER3 — Simultaneous Start Control Register 3

The PIC1ASSER3 register enables a start trigger for each channel of OSTMn.

Access: This register can be read/written in 16-bit units.

Address: FFDD 001C_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|--------------|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PIC1ASSER301 | PIC1ASSER300 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 29.15 PIC1ASSER3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 15 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | PIC1ASSER301 | Enables or disables a simultaneous start trigger for OSTM1. 0: Disabled 1: Enabled |
| 0 | PIC1ASSER300 | Enables or disables a simultaneous start trigger for OSTM0. 0: Disabled 1: Enabled |

29.2.2.7 PIC1AINn0 — Flip-Flop Circuit Initialization Register n0

The PIC1AINn0 register enables initialization of the RS flip-flop circuits 4 to 2 (RSn4 to 2).

Access: This register can be read/written in 8-bit units.

Address: FFDD 0020_H (n=0), FFDD 002C_H (n=1)

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|------------|------------|------------|---|---|
| | — | — | — | PIC1AINn04 | PIC1AINn03 | PIC1AINn02 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R | R |

Table 29.16 PIC1AINn0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 2 | PIC1AINn0[4:2] | Enables or disables initialization of the RS flip-flop circuits 4 to 2 (RSn4 to RSn2) used for the PWM output function with dead time. These bits are always read as 0. 0: Disabled 1: Initialized |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

29.2.2.8 PIC1AINn1 — DT Initialization Register n1

The PIC1AINn1 register enables initialization of the latch & toggle (DT) circuit.

Access: This register can be read/written in 8-bit units.

Address: FFDD 0024_H (n=0), FFDD 0030_H (n=1)

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|------------|------------|------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | PIC1AINn12 | PIC1AINn11 | PIC1AINn10 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 29.17 PIC1AINn1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 7 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | PIC1AINn1[2:0] | Enables or disables initialization of the DT circuit to be used for the trigger pulse width measurement function. These bits are always read as 0. 0: Disabled 1: Initialized |

29.2.2.9 PIC1ATSGHALLSEL — Hall Sensor Input Select Register

The PIC1ATSGHALLSEL register sets the pin conditions to input the external hall sensor signal.

Access: This register can be read/written in 8-bit units.

Address: FFDD 0074_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|-------------|-------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | TSG1HALLSEL | TSG0HALLSEL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 29.18 PIC1ATSGHALLSEL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | TSG1HALLSEL | Sets pin condition to input the external hall sensor signal.* ¹ 0: Separate input 1: Alternative input with ENCA |
| 0 | TSG0HALLSEL | Sets pin condition to input the external hall sensor signal.* ¹ 0: Separate input 1: Alternative input with ENCA |

Note 1. When TSG3nPTS12 to TSG3nPTS10 are used for input pattern to the TSG3, always set this bit to 1. In addition, set bit0 in the PIC1AREG50 register and bit0 in the PIC1AREG51 register as described below. When those bits are not used, do not change the setting from the value after reset.

| TSG1HALLSEL | PIC1AREG5100 | Function |
|------------------|--------------|---|
| 1 | 1 | Select input pin ENCA1E0, ENCA1E1, and ENCA1EC. |
| Other than above | | Setting prohibited. |

| TSG0HALLSEL | PIC1AREG5000 | Function |
|------------------|--------------|---|
| 1 | 0 | Select input pin ENCA0E0, ENCA0E1 and ENCA0EC |
| Other than above | | Setting prohibited |

29.2.2.10 PIC1ATAUD0SEL — TAUD0 Input Select Register

The PIC1ATAUD0SEL register is a 32-bit register that selects TAUDTIN input signals.

Access: This register can be read/written in 32-bit units.

Address: FFDD 0078_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|
| | PIC1ATAUD0IN143 | PIC1ATAUD0IN142 | PIC1ATAUD0IN141 | PIC1ATAUD0IN140 | PIC1ATAUD0IN123 | PIC1ATAUD0IN122 | PIC1ATAUD0IN121 | PIC1ATAUD0IN120 | PIC1ATAUD0IN103 | PIC1ATAUD0IN102 | PIC1ATAUD0IN101 | PIC1ATAUD0IN100 | PIC1ATAUD0IN83 | PIC1ATAUD0IN82 | PIC1ATAUD0IN81 | PIC1ATAUD0IN80 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PIC1ATAUD0IN63 | PIC1ATAUD0IN62 | PIC1ATAUD0IN61 | PIC1ATAUD0IN60 | PIC1ATAUD0IN43 | PIC1ATAUD0IN42 | PIC1ATAUD0IN41 | PIC1ATAUD0IN40 | PIC1ATAUD0IN23 | PIC1ATAUD0IN22 | PIC1ATAUD0IN21 | PIC1ATAUD0IN20 | PIC1ATAUD0IN03 | PIC1ATAUD0IN02 | PIC1ATAUD0IN01 | PIC1ATAUD0IN00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.19 PIC1ATAUD0SEL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|--|
| 2m+3 2m+2 | PIC1ATAUD0IN m[3:2] | Selects the signal to output to TAUD0TIN (m+1) output pin. 00: TAUD0TIN (m+1) is selected. 01: TAUD0TIN (m) is selected. 10: TAUD1TIN (m+1) is selected. 11: TAUD1TIN (m) is selected. |
| 2m+1 2m | PIC1ATAUD0IN m[1:0] | Selects the signal to output to TAUD0TIN (m) output pin. 00: TAUD0TIN (m) is selected. 01: TAUD0TIN (m+1) is selected. 10: TAUD1TIN (m) is selected. 11: TAUD1TIN (m+1) is selected. |

Note: m is an even channel number of TAUD0 (CHm_even)

29.2.2.11 PIC1ATAUD1SEL — TAUD1 Input Select Register

The PIC1ATAUD1SEL register is a 32-bit register that selects TAUDTIN input signals.

Access: This register can be read/written in 32-bit units.

Address: FFDD 007C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|----------------|
| | PIC1ATAUD1IN143 | PIC1ATAUD1IN142 | PIC1ATAUD1IN141 | PIC1ATAUD1IN140 | PIC1ATAUD1IN123 | PIC1ATAUD1IN122 | PIC1ATAUD1IN121 | PIC1ATAUD1IN120 | PIC1ATAUD1IN103 | PIC1ATAUD1IN102 | PIC1ATAUD1IN101 | PIC1ATAUD1IN100 | PIC1ATAUD1IN83 | PIC1ATAUD1IN82 | PIC1ATAUD1IN81 | PIC1ATAUD1IN80 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PIC1ATAUD1IN63 | PIC1ATAUD1IN62 | PIC1ATAUD1IN61 | PIC1ATAUD1IN60 | PIC1ATAUD1IN43 | PIC1ATAUD1IN42 | PIC1ATAUD1IN41 | PIC1ATAUD1IN40 | PIC1ATAUD1IN23 | PIC1ATAUD1IN22 | PIC1ATAUD1IN21 | PIC1ATAUD1IN20 | PIC1ATAUD1IN03 | PIC1ATAUD1IN02 | PIC1ATAUD1IN01 | PIC1ATAUD1IN00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.20 PIC1ATAUD1SEL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 2m+3 2m+2 | PIC1ATAUD1IN m[3:2] | Selects the signal to output to TAUD1TIN (m+1) output pin 00: TAUD1TIN (m+1) is selected. 01: TAUD1TIN (m) is selected. 10: TAUD0TIN (m+1) is selected. 11: TAUD0TIN (m) is selected. |
| 2m+1 2m | PIC1ATAUD1IN m[1:0] | Selects the signal to output to TAUD1TIN (m) output pin 00: TAUD1TIN (m) is selected. 01: TAUD1TIN (m+1) is selected. 10: TAUD0TIN (m) is selected. 11: TAUD0TIN (m+1) is selected. |

Note: m is an even channel number of TAUD1 (CH_{m_even})

29.2.2.12 PIC1AHIZCEN0 — Hi-Z Control Register 0

The PIC1AHIZCEN0 register selects Hi-Z control input signals of TAUD0.

Access: This register can be read/written in 8-bit units.

Address: FFDD 0080_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------------------|-------------------|-------------------|---|---|---|---|-------------------|
| | PIC1AHIZCEN 07 | PIC1AHIZCEN 06 | PIC1AHIZCEN 05 | — | — | — | — | PIC1AHIZCEN 00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R/W |

Table 29.21 PIC1AHIZCEN0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 7 | PIC1AHIZCEN 07 | Enables or disables Hi-Z control by AD CG1 error interrupt (INTADCG1ERR). 0: Disabled 1: Enabled |
| 6 | PIC1AHIZCEN 06 | Enables or disables Hi-Z control by AD CG0 error interrupt (INTADCG0ERR). 0: Disabled 1: Enabled |
| 5 | PIC1AHIZCEN 05 | Enables or disables Hi-Z control by ERROROUT signal. 0: Disabled 1: Enabled |
| 4 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PIC1AHIZCEN 00 | Enables or disables Hi-Z control by TAPA0ESO pin input. 0: Disabled 1: Enabled |

CAUTIONS

1. Set this register before starting U/V/W outputs or UB/VB/WB outputs of TAUD0.
2. Set TAPA0CTL0.TAPA0DCN = 0 and TAPA0CTL0.TAPA0DCP = 1 when performing Hi-Z control by AD CG error signal and ERROROUT signal.

29.2.2.13 PIC1AHIZCEN1 — Hi-Z Control Register 1

The PIC1AHIZCEN1 register selects Hi-Z control input signals of TAUD1.

Access: This register can be read/written in 8-bit units.

Address: FFDD 0084_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------------------|-------------------|-------------------|---|---|---|---|-------------------|
| | PIC1AHIZCEN 17 | PIC1AHIZCEN 16 | PIC1AHIZCEN 15 | — | — | — | — | PIC1AHIZCEN 10 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R/W |

Table 29.22 PIC1AHIZCEN1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 7 | PIC1AHIZCEN 17 | Enables or disables Hi-Z control by AD CG1 error interrupt (INTADCG1ERR). 0: Disabled 1: Enabled |
| 6 | PIC1AHIZCEN 16 | Enables or disables Hi-Z control by AD CG0 error interrupt (INTADCG0ERR). 0: Disabled 1: Enabled |
| 5 | PIC1AHIZCEN 15 | Enables or disables Hi-Z control by ERROROUT signal. 0: Disabled 1: Enabled |
| 4 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PIC1AHIZCEN 10 | Enables or disables Hi-Z control by TAPA1ESO pin input. 0: Disabled 1: Enabled |

CAUTIONS

1. Set this register before starting U/V/W outputs or UB/VB/WB outputs of TAUD1.
2. Set TAPA1CTL0.TAPA1DCN = 0 and TAPA1CTL0.TAPA1DCP = 1 when performing Hi-Z control by AD CG error signal and ERROROUT signal.

29.2.2.14 PIC1AHIZCEN2 — Hi-Z Control Register 2

The PIC1AHIZCEN2 register selects Hi-Z control input signals of TSG30.

Access: This register can be read/written in 8-bit units.

Address: FFDD 0088_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------------------|-------------------|-------------------|---|-------------------|---|---|-------------------|
| | PIC1AHIZCEN 27 | PIC1AHIZCEN 26 | PIC1AHIZCEN 25 | — | PIC1AHIZCEN 23 | — | — | PIC1AHIZCEN 20 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R/W | R | R | R/W |

Table 29.23 PIC1AHIZCEN2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 7 | PIC1AHIZCEN 27 | Enables or disables Hi-Z control by AD CG1 error interrupt (INTADCG1ERR). 0: Disabled 1: Enabled |
| 6 | PIC1AHIZCEN 26 | Enables or disables Hi-Z control by AD CG0 error interrupt (INTADCG0ERR). 0: Disabled 1: Enabled |
| 5 | PIC1AHIZCEN 25 | Enables or disables Hi-Z control by ERROROUT signal. 0: Disabled 1: Enabled |
| 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | PIC1AHIZCEN 23 | Enables or disables Hi-Z control by INTTSG30IER interrupt signal. 0: Disabled 1: Enabled |
| 2, 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PIC1AHIZCEN 20 | Enables or disables Hi-Z control by TAPA0ESO pin input. 0: Disabled 1: Enabled |

CAUTIONS

1. Set this register before starting TSG30 output.
2. Set TAPA2CTL0.TAPA2DCN = 0 and TAPA2CTL0.TAPA2DCP = 1 when performing Hi-Z control by AD CG error signal, ERROROUT signal, and TSG30 error signal.

29.2.2.15 PIC1AHIZCEN3 — Hi-Z Control Register 3

The PIC1AHIZCEN3 register selects Hi-Z control input signals of TSG31.

Access: This register can be read/written in 8-bit units.

Address: FFDD 008C_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|-------------------|-------------------|-------------------|-------------------|---|---|---|-------------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PIC1AHIZCEN 37 | PIC1AHIZCEN 36 | PIC1AHIZCEN 35 | PIC1AHIZCEN 34 | — | — | — | PIC1AHIZCEN 30 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R | R/W |

Table 29.24 PIC1AHIZCEN3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 7 | PIC1AHIZCEN 37 | Enables or disables Hi-Z control by AD CG1 error interrupt (INTADCG1ERR) 0: Disabled 1: Enabled |
| 6 | PIC1AHIZCEN 36 | Enables or disables Hi-Z control by AD CG0 error interrupt (INTADCG0ERR). 0: Disabled 1: Enabled |
| 5 | PIC1AHIZCEN 35 | Enables or disables Hi-Z control by ERROROUT signal 0: Disabled 1: Enabled |
| 4 | PIC1AHIZCEN 34 | Enables or disables Hi-Z control by INTTSG31IER interrupt signal. 0: Disabled 1: Enabled |
| 3 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PIC1AHIZCEN 30 | Enables or disables Hi-Z control by TAPA1ESO pin input. 0: Disabled 1: Enabled |

CAUTION

Set this register before starting TSG31 output.

Set TAPA3CTL0.TAPA3DCN = 0 and TAPA3CTL0.TAPA3DCP = 1 when performing Hi-Z control by AD CG error signal, ERROROUT signal, and TSG31 error signal.

29.2.2.16 PIC1AENCSEL400 — ENCATIN1 Input Select Register 400

The PIC1AENCSEL400 register is used for encoder capture trigger select function.

Access: This register can be read/written in 8-bit units.

Address: FFDD 00B8_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------------|---|---|---|---------------------|---------------------|---------------------|---------------------|
| | PIC1AENCSEL 4007 | — | — | — | PIC1AENCSEL 4003 | PIC1AENCSEL 4002 | PIC1AENCSEL 4001 | PIC1AENCSEL 4000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R/W | R/W | R/W | R/W |

Table 29.25 PIC1AENSEL400 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|--|
| 7 | PIC1AENCSEL 4007 | Enables or disables output of INTTAUD0Im signal selected by PIC1AENCSEL400[3:0]. 0: Disabled 1: Enabled |
| 6 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | PIC1AENCSEL 400[3:0] | Selects TAUD0TINTm to be used as a capture trigger signal for ENCA0 and ENCA1. 0000: INTTAUD0I0 is selected 0001: INTTAUD0I1 is selected 0010: INTTAUD0I2 is selected 0011: INTTAUD0I3 is selected 0100: INTTAUD0I4 is selected 0101: INTTAUD0I5 is selected 0110: INTTAUD0I6 is selected 0111: INTTAUD0I7 is selected 1000: INTTAUD0I8 is selected 1001: INTTAUD0I9 is selected 1010: INTTAUD0I10 is selected 1011: INTTAUD0I11 is selected 1100: INTTAUD0I12 is selected 1101: INTTAUD0I13 is selected 1110: INTTAUD0I14 is selected 1111: INTTAUD0I15 is selected |

29.2.2.17 PIC1AENCSEL410 — ENCATIN1 Input Select Register 410

The PIC1AENCSEL410 register is used for encoder capture trigger select function.

Access: This register can be read/written in 8-bit units.

Address: FFDD 00BC_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------------|---|---|---|---------------------|---------------------|---------------------|---------------------|
| | PIC1AENCSEL 4107 | — | — | — | PIC1AENCSEL 4103 | PIC1AENCSEL 4102 | PIC1AENCSEL 4101 | PIC1AENCSEL 4100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R/W | R/W | R/W | R/W |

Table 29.26 PIC1AENCSEL410 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|--|
| 7 | PIC1AENCSEL 4107 | Enables or disables output of INTTAUD1 _m signal selected by PIC1AENCSEL410[3:0]. 0: Disabled 1: Enabled |
| 6 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | PIC1AENCSEL 410[3:0] | Selects TAUD1TINT _m to be used as a capture trigger signal for ENCA0 and ENCA1. 0000: INTTAUD110 is selected 0001: INTTAUD111 is selected 0010: INTTAUD112 is selected 0011: INTTAUD113 is selected 0100: INTTAUD114 is selected 0101: INTTAUD115 is selected 0110: INTTAUD116 is selected 0111: INTTAUD117 is selected 1000: INTTAUD118 is selected 1001: INTTAUD119 is selected 1010: INTTAUD1110 is selected 1011: INTTAUD1111 is selected 1100: INTTAUD1112 is selected 1101: INTTAUD1113 is selected 1110: INTTAUD1114 is selected 1111: INTTAUD1115 is selected |

29.2.2.18 PIC1AREG200 — Timer Input/Output Control Register 200

The PIC1AREG200 register selects TAUD0 input signals.

Access: This register can be read/written in 32-bit units.

Address: FFDD 00C0_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|-----------------------|-----------------------|-----------------------|-----------------------|----|----|----|----|-----------------------|-----------------------|-----------------------|-----------------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | PIC1AR EG200 18 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | PIC1AR EG200 11 | PIC1AR EG200 10 | PIC1AR EG200 09 | PIC1AR EG200 08 | — | — | — | — | PIC1AR EG200 03 | PIC1AR EG200 02 | PIC1AR EG200 01 | PIC1AR EG200 00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 29.27 PIC1AREG200 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|--|
| 31 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PIC1AREG20018 | Selects the signal for input as TAUD0TIN10, TAUD0TIN12, and TAUD0TIN14 signals of TAUD0. 1: Select TOUT of TAUD0 CH02. Settings other than above are prohibited.*1 |
| 17 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11, 10 | PIC1AREG200 [11:10] | Selects the signal to be input as TAUD0TIN6, TAUD0TIN7 signals of TAUD0. 10: TS0PTE signal of TSG3. Settings other than above are prohibited.*1 |
| 9, 8 | PIC1AREG200 [09:08] | Selects the signal to be input as TAUD0TIN4, TAUD0TIN5 signals of TAUD0. 10: TS0PTE signal of TSG3. Settings other than above are prohibited.*1 |
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | PIC1AREG20003 | Selects the signal to be input as TAUD0TIN7 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1AREG20011, PIC1AREG20010 bits (TS0PTE signal). |
| 2 | PIC1AREG20002 | Selects the signal to be input as TAUD0TIN6 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1AREG20011 and PIC1AREG20010 bits (TS0PTE signal). |
| 1 | PIC1AREG20001 | Selects the signal to be input as TAUD0TIN5 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1AREG20009, PIC1AREG20008 bits (TS0PTE signal). |
| 0 | PIC1AREG20000 | Selects the signal to be input as TAUD0TIN4 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1AREG20009 and PIC1AREG20008 bits (TS0PTE signal). |

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used.

29.2.2.19 PIC1AREG210 — Timer Input/Output Control Register 210

The PIC1AREG210 register selects TAUD1 input signals.

Access: This register can be read/written in 32-bit units.

Address: FFDD 00D4_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|-----------------------|-----------------------|-----------------------|-----------------------|----|----|----|----|-----------------------|-----------------------|-----------------------|-----------------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | PIC1AR EG210 18 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | PIC1AR EG210 11 | PIC1AR EG210 10 | PIC1AR EG210 09 | PIC1AR EG210 08 | — | — | — | — | PIC1AR EG210 03 | PIC1AR EG210 02 | PIC1AR EG210 01 | PIC1AR EG210 00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 29.28 PIC1AREG210 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|--|
| 31 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PIC1AREG21018 | Selects the signal for input as TAUD1TIN10, TAUD1TIN12, and TAUD1TIN14 signals of TAUD1. 1: Select TOUT of TAUD1 CH02. Settings other than above are prohibited.*1 |
| 17 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11, 10 | PIC1AREG210 [11:10] | Selects the signal to be input as TAUD1TIN6 and TAUD1TIN7 signals of TAUD1. 10: TS0PTE signal of TSG31 Settings other than above are prohibited.*1 |
| 9, 8 | PIC1AREG210 [09:08] | Selects the signal to be input as TAUD1TIN4 and TAUD1TIN5 signals of TAUD1. 10: TS0PTE signal of TSG31 Settings other than above are prohibited.*1 |
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | PIC1AREG21003 | Selects the signal to be input as TAUD1TIN7 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1AREG21011 and PIC1AREG21010 bits (TS0PTE signal). |
| 2 | PIC1AREG21002 | Selects the signal to be input as TAUD1TIN6 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1AREG21011 and PIC1AREG21010 bits (TS0PTE signal). |
| 1 | PIC1AREG21001 | Selects the signal to be input as TAUD1TIN5 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1AREG21009 and PIC1AREG21008 bits (TS0PTE signal). |
| 0 | PIC1AREG21000 | Selects the signal to be input as TAUD1TIN4 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1AREG21009 and PIC1AREG21008 bits (TS0PTE signal). |

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the

value after reset when not used.

29.2.2.20 PIC1AREG2n1 — Timer Input/Output Control Register 2n1

The PIC1AREG2n1 register selects the logical operation for the combination circuit PFN0xx.

Access: This register can be read/written in 32-bit units.

Address: FFDD 00C4_H(n = 0), FFDD 00D8_H(n = 1)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | PIC1AREG2n1 27 | PIC1AREG2n1 26 | PIC1AREG2n1 25 | PIC1AREG2n1 24 | PIC1AREG2n1 23 | PIC1AREG2n1 22 | PIC1AREG2n1 21 | PIC1AREG2n1 20 | PIC1AREG2n1 19 | PIC1AREG2n1 18 | PIC1AREG2n1 17 | PIC1AREG2n1 16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 29.29 PIC1AREG2n1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 31 to 28 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 27, 26 | PIC1AREG2n1 [27:26] | Selects PFN045 WO2 output.* ¹ 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.* ² |
| 25, 24 | PIC1AREG2n1 [25:24] | Selects PFN045 WO1 output.* ¹ 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.* ² |
| 23, 22 | PIC1AREG2n1 [23:22] | Selects PFCN023 VO2 output.* ¹ 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.* ² |
| 21, 20 | PIC1AREG2n1 [21:20] | Selects PFN023 VO1 output.* ¹ 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.* ² |
| 19, 18 | PIC1AREG2n1 [19:18] | Selects PFN001 UO2 output.* ¹ 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.* ² |
| 17, 16 | PIC1AREG2n1 [17:16] | Selects PFN001 UO1 output.* ¹ 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.* ² |
| 15 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Note 1. The register value for some functions needs to be set depending on the value of TAUD. For the setting values, see **Section 29.2.3, Function**.

Note 2. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used.

Block diagram of PFN001 is shown in the following figure.

PFN023 and PFN045 are operated under the same logic with different input signals and select registers.

For connection of PFN0xx with peripheral circuits, see **Figure 29.11**.

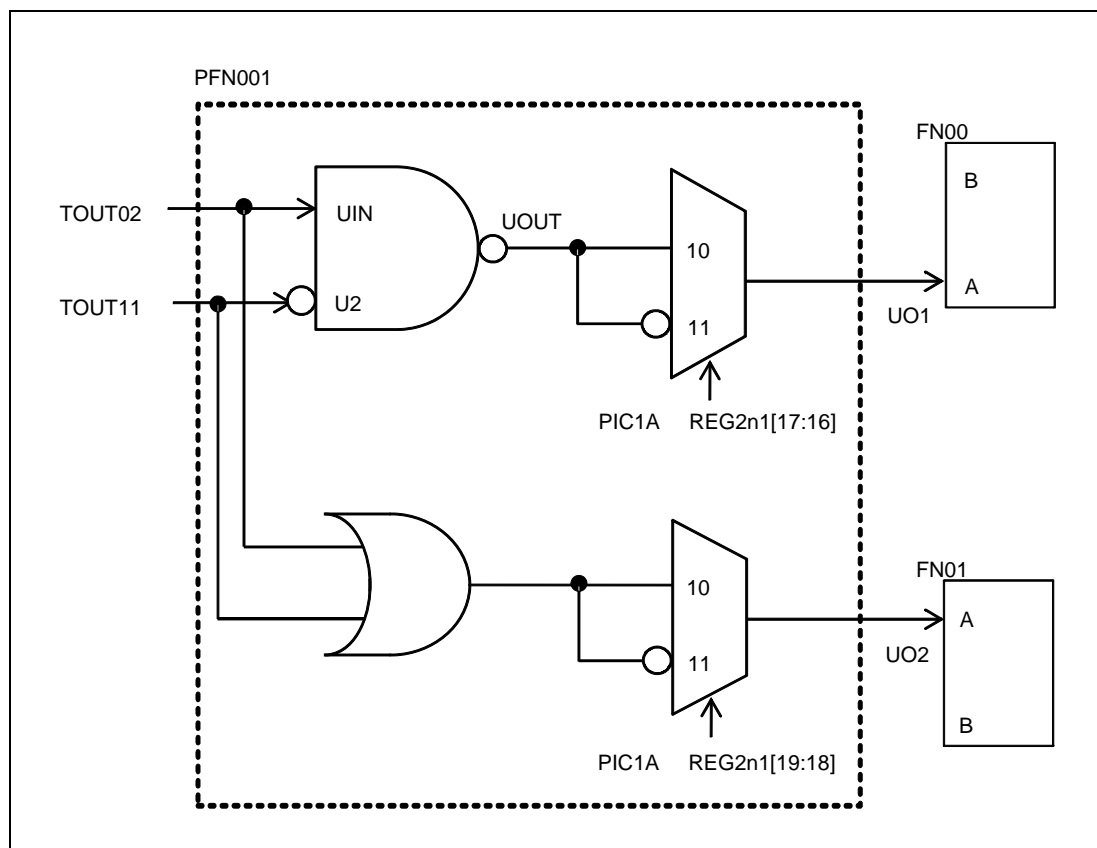


Figure 29.1 Block Diagram of PFN001

29.2.2.21 PIC1AREG2n2 — Timer Input/Output Control Register 2n2

The PIC1AREG2n2 register selects input signals of TAUDn CHm.

Access: This register can be read/written in 32-bit units.

Address: FFDD 00C8_H (n=0), FFDD 00DC_H (n=1)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| | — | — | — | — | PIC1AREG2n2 27 | PIC1AREG2n2 26 | PIC1AREG2n2 25 | PIC1AREG2n2 24 | PIC1AREG2n2 23 | PIC1AREG2n2 22 | PIC1AREG2n2 21 | PIC1AREG2n2 20 | PIC1AREG2n2 19 | PIC1AREG2n2 18 | PIC1AREG2n2 17 | PIC1AREG2n2 16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | PIC1AREG2n2 04 | PIC1AREG2n2 03 | PIC1AREG2n2 02 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R | R |

Table 29.30 PIC1AREG2n2 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------------------|--|
| 31 to 28 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 27, 26 | PIC1AREG2n2 [27:26] | Selects the TIN input signal of TAUDnCH15. 00: TIN pin input. 10: Signal selected by PIC1AREG2n204 bit (TOUT of TAUDnCH09). Settings other than above are prohibited. |
| 25, 24 | PIC1AREG2n2 [25:24] | Selects the TIN input signal of TAUDnCH14. 00: TIN pin input 10: Signal selected by PIC1AREG2n018 register bit (TOUT of TAUDnCH02). Settings other than above are prohibited. |
| 23, 22 | PIC1AREG2n2 [23:22] | Selects the TIN input signal of TAUDnCH13. 00: TIN pin input 10: Signal selected by PIC1AREG2n203 bit (TOUT of TAUDnCH07). Settings other than above are prohibited. |
| 21, 20 | PIC1AREG2n2 [21:20] | Selects the TIN input signal of TAUDnCH12. 00: TIN pin input 10: Signal selected by PIC1AREG2n018 register bit (TOUT of TAUDnCH02). Settings other than above are prohibited. |
| 19, 18 | PIC1AREG2n2 [19:18] | Selects the TIN input signal of TAUDnCH11. 00: TIN pin input 10: Signal selected by PIC1AREG2n202 bit (TOUT of TAUDnCH05). Settings other than above are prohibited. |
| 17, 16 | PIC1AREG2n2 [17:16] | Selects the TIN input signal of TAUDnCH10. 00: TIN pin input 10: Signal selected by PIC1AREG2n018 register bit (TOUT of TAUDnCH02). Settings other than above are prohibited. |
| 15 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | PIC1AREG2n2 04 | Selects the signal to be supplied to TIN of TAUDnCH15. 0: TOUT of TAUDnCH09. 1: Set/clear output by INTTAUDnI8 and INTTAUDnI9. |
| 3 | PIC1AREG2n2 03 | Selects the signal to be supplied to TIN of TAUDnCH13. 0: TOUT of TAUDnCH07. 1: Set/clear output by INTTAUDnI6 and INTTAUDnI7. |

Table 29.30 PIC1AREG2n2 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 2 | PIC1AREG2n2 02 | Selects the signal to be supplied to TIN of TAUDnCH11. 0: TOUT of TAUDnCH05. 1: Set/clear output by INTTAUDnI4 and INTTAUDnI5. |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

29.2.2.22 PIC1AREG2n3 — Timer Input/Output Control Register 2n3

The PIC1AREG2n3 register selects logical operation for the combination circuit FN0i.

Access: This register can be read/written in 32-bit units.

Address: FFDD 00CC_H (n = 0), FFDD 00E0_H (n = 1)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|-----------------------|-----------------------|-----------------------|----|-----------------------|-----------------------|-----------------------|----|-----------------------|-----------------------|-----------------------|----|-----------------------|-----------------------|-----------------------|
| | — | — | — | — | — | — | — | — | — | PIC1AR EG2n3 22 | PIC1AR EG2n3 21 | PIC1AR EG2n3 20 | — | PIC1AR EG2n3 18 | PIC1AR EG2n3 17 | PIC1AR EG2n3 16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | PIC1AR EG2n3 14 | PIC1AR EG2n3 13 | PIC1AR EG2n3 12 | — | PIC1AR EG2n3 10 | PIC1AR EG2n3 09 | PIC1AR EG2n3 08 | — | PIC1AR EG2n3 06 | PIC1AR EG2n3 05 | PIC1AR EG2n3 04 | — | PIC1AR EG2n3 02 | PIC1AR EG2n3 01 | PIC1AR EG2n3 00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W | R | R/W | R/W | R/W |

Table 29.31 PIC1AREG2n3 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 to 20 | PIC1AREG2n3 [22:20] | Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited. |
| 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 to 16 | PIC1AREG2n3 [18:16] | Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited. |
| 15 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 14 to 12 | PIC1AREG2n3 [14:12] | Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited. |
| 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 29.31 PIC1AREG2n3 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 10 to 8 | PIC1AREG2n3 [10:08] | Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited. |
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 4 | PIC1AREG2n3 [06:04] | Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited. |
| 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | PIC1AREG2n3 [02:00] | Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited. |

Note 1. The register value for some functions needs to be set according to the value of TAUD. For the setting values, see **Section 29.2.3, Function**.

Block diagram of FN00 is shown in the following figure.

FN01 to FN05 are operated under the same logic with different input signals and select registers.

For connection of FN0i with peripheral circuits, see **Figure 29.11**.

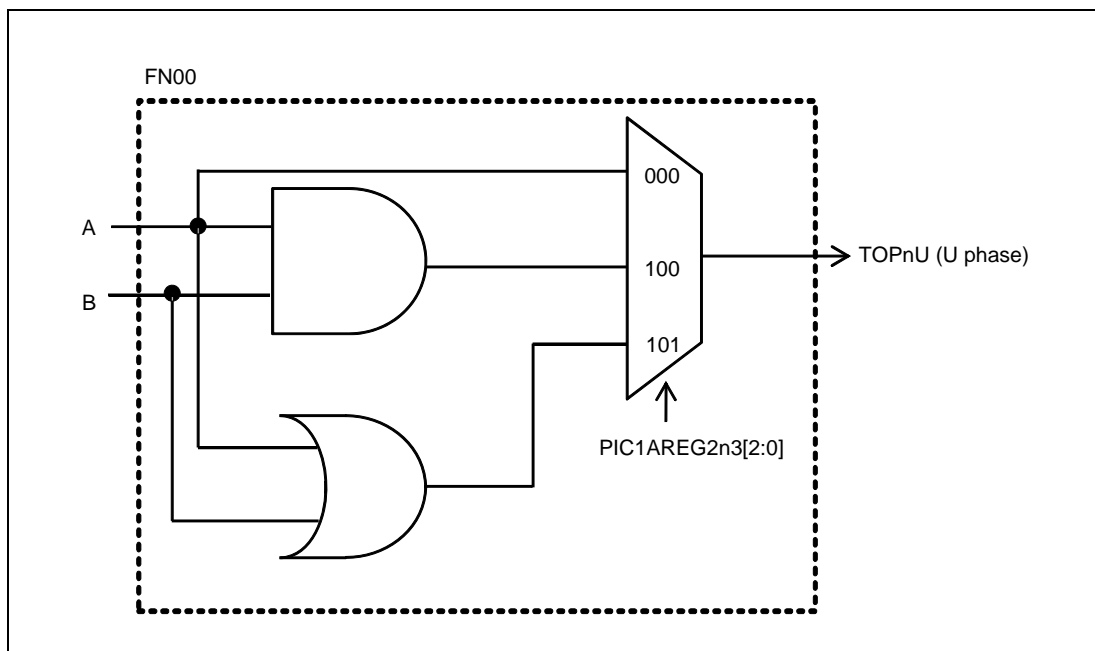


Figure 29.2 Block Diagram of FN00

29.2.2.23 PIC1AREG30 — Timer Input/Output Control Register 30

The PIC1AREG30 register selects ENCA_n input signals.

Access: This register can be read/written in 32-bit units.

Address: FFDD 00E8_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | PIC1AREG3022 | PIC1AREG3021 | PIC1AREG3020 | PIC1AREG3019 | PIC1AREG3018 | PIC1AREG3017 | PIC1AREG3016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PIC1AREG3015 | PIC1AREG3014 | PIC1AREG3013 | PIC1AREG3012 | PIC1AREG3011 | PIC1AREG3010 | PIC1AREG3009 | PIC1AREG3008 | PIC1AREG3007 | PIC1AREG3006 | PIC1AREG3005 | PIC1AREG3004 | PIC1AREG3003 | PIC1AREG3002 | PIC1AREG3001 | PIC1AREG3000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.32 PIC1AREG30 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|--------------------|--|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22 | PIC1AREG3022 | Selects the input pins (ENCA0E0, ENCA0E1, ENCA0EC internal inputs) of ENCA0 timer. 0: The signal selected by PIC1AREG3000 (ENCA0E0), PIC1AREG3001 (ENCA0E1), and PIC1AREG3017-16 (ENCA0EC) 1: The signal selected by PIC1AREG3020 and PIC1AREG3019 |
| 21 | PIC1AREG3021 | Selects the signal to be supplied to PIC1AREG3012-15. 0: ENCA111 (signal 1 of the ENCA1 external pin) 1: The signal selected by the PIC1AENCSEL4107 bit of the PIC1AENCSEL410 register. |
| 20,19 | PIC1AREG30 [20:19] | Selects the input pins (ENCA1E0, ENCA1E1, ENCA1EC internal inputs) of ENCA1 timer. 00: The ENCA1E0, ENCA1E1, ENCA1EC input pins of ENCA1 timer. Settings other than above are prohibited. |
| 18 | PIC1AREG3018 | Selects the signal to be supplied to PIC1AREG3002 to 05. 0: ENCA011 (signal 1 of ENCA0 external pin) 1: The signal selected by the PIC1AENCSEL4007 bit of the PIC1AENCSEL400 register. |
| 17, 16 | PIC1AREG30 [17:16] | Selects the input pins (ENCA0E0, ENCA0E1, ENCA0EC internal inputs) of ENCA0 timer. 00: The ENCA0E0, ENCA0E1, ENCA0EC input pins of ENCA0 timer. Settings other than above are prohibited. |
| 15 to 12 | PIC1AREG30 [15:12] | Selects the signal to be input as the ENCA1TIN1 signal. 0: The signal selected by PIC1AREG3021. 1: The signal selected by PIC1AREG3018. 2: ADCG0TRG4 3: ADCG0TRG3 4: ADCG0TRG2 5: ADCG0TRG1 6: ADCG0TRG0 7: ADCG1TRG4 8: ADCG1TRG3 9: ADCG1TRG2 10: ADCG1TRG1 11: ADCG1TRG0 Settings other than above are prohibited. |

Table 29.32 PIC1AREG30 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 11, 10 | PIC1AREG30 [11:10] | Selects the ENCAEC pin input of timer ENCA1. 00: The signal selected by the PIC1AREG3019 and PIC1AREG3020 bits. 10: The signal selected by the PIC1AREG3016 and PIC1AREG3017 bits. 11: ENCA0EQ1 signal (ENCA0 timer) Settings other than above are prohibited |
| 9, 8 | PIC1AREG30 [09:08] | Selects the ENCA1E1 pin input of timer ENCA1. 00: Signal selected by the PIC1AREG3019 and PIC1AREG3020 bits. 01: The signal selected by the PIC1AREG3016 and PIC1AREG3017 bits. 10: TS1PUD signal of TSG31. Settings other than above are prohibited. |
| 7, 6 | PIC1AREG30 [07:06] | Selects the ENCA1E0 pin input of timer ENCA1. 00: The signal selected by the PIC1AREG3019 and PIC1AREG3020 bits. 01: The signal selected by the PIC1AREG3016 and PIC1AREG3017 bits. 10: TS1PEC signal of TSG31. Settings other than above are prohibited |
| 5 to 2 | PIC1AREG30 [05:02] | Selects the signal to be input as the ENCAT0TIN1 signal. 0: The signal selected by the PIC1AREG3018 bit. 1: The signal selected by the PIC1AREG3021 bit. 2: ADCG0TRG4 3: ADCG0TRG3 4: ADCG0TRG2 5: ADCG0TRG1 6: ADCG0TRG0 7: ADCG1TRG4 8: ADCG1TRG3 9: ADCG1TRG2 10: ADCG1TRG1 11 : ADCG1TRG0 Settings other than above are prohibited. |
| 1 | PIC1AREG3001 | Selects the signal to input as the ENCA0E1 internal signal. 0: Signal selected by PIC1AREG3017 and PIC1AREG3016. 1: TS0PUD signal of TSG30. |
| 0 | PIC1AREG3000 | Selects the signal to input as the ENCA0E0 internal signal. 0: Signal selected by PIC1AREG3017 and PIC1AREG3016. 1: TS0PEC signal of TSG30. |

29.2.2.24 PIC1AREG31 — Timer Input/Output Control Register 31

The PIC1AREG31 register selects TAUDn and TAUJ0 input signals.

Access: This register can be read/written in 32-bit units.

Address: FFDD 00EC_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------------|----|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | — | — | — | — | — | — | — | — | — | PIC1AR EG3122 | PIC1AR EG3121 | PIC1AR EG3120 | PIC1AR EG3119 | PIC1AR EG3118 | PIC1AR EG3117 | PIC1AR EG3116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PIC1AR EG3115 | — | PIC1AR EG3113 | PIC1AR EG3112 | PIC1AR EG3111 | PIC1AR EG3110 | PIC1AR EG3109 | PIC1AR EG3108 | PIC1AR EG3107 | PIC1AR EG3106 | — | PIC1AR EG3104 | PIC1AR EG3103 | — | PIC1AR EG3101 | PIC1AR EG3100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | R/W | R | R/W | R/W |

Table 29.33 PIC1AREG31 Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------------------|--|
| 31 to 23 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 22, 21 | PIC1AREG31 [22:21] | Selects the TIN input signal of TAUD1CH02. 00: TIN pin input 01: DT output signal of ENCAT1EQ0 Settings other than above are prohibited. |
| 20 | PIC1AREG3120 | Selects the TIN input signal of TAUD1CH01. 0: TIN pin input 1: The signal selected by PIC1AREG3115 to PIC1AREG3117. |
| 19, 18 | PIC1AREG31 [19:18] | Selects the TIN input signal of TAUD1CH00. 00: The signal selected by PIC1AREG3115 to PIC1AREG3117. 10: DT output signal of ENCAT1EQ0 Settings other than above are prohibited. |
| 17 to 15 | PIC1AREG31 [17:15] | Selects the TIN input signal of TAUD1CH00 and TAUD1CH01. 000: TIN pin input. 001: DT output signal of ENCAT1EQ1 Settings other than above are prohibited. |
| 14 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 13, 12 | PIC1AREG31 [13:12] | Selects the TIN input signal of TAUD0CH02. 00: TIN pin input. 10: DT output signal of ENCAT0EQ0 Settings other than above are prohibited. |
| 11 | PIC1AREG3111 | Selects the TIN input signal of TAUD0CH01. 0: TIN pin input. 1: The signal selected by PIC1AREG3106 to PIC1AREG3108. |
| 10, 9 | PIC1AREG31 [10:09] | Selects the TIN input signal of TAUD0CH00. 00: The signal selected by PIC1AREG3106 to PIC1AREG3108. 10: DT output signal of ENCAT0EQ0 Settings other than above are prohibited. |
| 8 to 6 | PIC1AREG31 [08:06] | Selects the TIN input signal of TAUD0CH00 and TAUD0CH01. 000: TIN pin input 001: DT output signal of ENCAT0EQ1 Settings other than above are prohibited. |

Table 29.33 PIC1AREG31 Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | PIC1AREG3104 | Selects the TIN input signal of TAUJ0CH03. 0: TIN pin input. 1: DT output signal of ENCAT1IEC. |
| 3 | PIC1AREG3103 | Selects the TIN input signal of TAUJ0CH02. 0: TIN pin input. 1: DT output signal of ENCAT1IEC |
| 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | PIC1AREG3101 | Selects the TIN input signal of TAUJ0CH01. 0: TIN pin input. 1: DT output signal of ENCAT0IEC |
| 0 | PIC1AREG3100 | Selects the TIN input signal of TAUJ0CH00. 0: TIN pin input. 1: DT output signal of ENCAT0IEC |

29.2.2.25 PIC1AREG50 — Timer Input/Output Control Register 50

The PIC1AREG50 register selects TSG30 input signals.

Access: This register can be read/written in 16-bit units.

Address: FFDD 00F8_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|--------------|---|--------------|--------------|--------------|--------------|---|---|---|---|--------------|
| | — | — | — | — | — | PIC1AREG5010 | — | PIC1AREG5008 | PIC1AREG5007 | PIC1AREG5006 | PIC1AREG5005 | — | — | — | — | PIC1AREG5000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R | R/W | R/W | R/W | R/W | R | R | R | R | R/W |

Table 29.34 PIC1AREG50 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 15 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | PIC1AREG5010 | Selects the signal to be input as the TSG30TSTOPC0 signal of the TSG30 timer. 0: INTENCA111 input of the ENCA1 timer 1: Setting prohibited. |
| 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | PIC1AREG5008 | Selects the signal to be input as the TSG30TSTOPC0 signal of the TSG30 timer. 0: INTENCA011 input of the ENCA0 timer 1: Setting prohibited. |
| 7 | PIC1AREG5007 | Selects the signal to be input as the TS0OPCI1 signal of the TSG30 timer. 0: INTTAUD017 input of the TAUD 1: Setting prohibited. |
| 6, 5 | PIC1AREG50 [06:05] | Selects the signal to be input as the TSG30TSTOPS (TS0OPCI0) signal of the TSG30 timer. 01: The signal selected by the PIC1AREG5008 bit 10: The signal selected by the PIC1AREG5010 bit 11: INTTAUD015 output signal of TAUD0. Settings other than above are prohibited. ¹ |
| 4 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PIC1AREG5000 | Switches the ENCA signal and external hall sensor signal. For the note on selection of the signals, see Note 1 of Section 29.2.2.9, PIC1ATSGHALLSEL — Hall Sensor Input Select Register . 0: Select the pin input ENCA0E0, ENCA0E1, ENCA0EC. 1: Setting prohibited. |

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used

29.2.2.26 PIC1AREG51 — Timer Input/Output Control Register 51

The PIC1AREG51 register selects TSG31 input signals.

Access: This register can be read/written in 16-bit units.

Address: FFDD 00FC_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|--------------|---|--------------|--------------|--------------|--------------|---|---|---|---|--------------|
| | — | — | — | — | — | PIC1AREG5110 | — | PIC1AREG5108 | PIC1AREG5107 | PIC1AREG5106 | PIC1AREG5105 | — | — | — | — | PIC1AREG5100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R | R/W | R/W | R/W | R/W | R | R | R | R | R/W |

Table 29.35 PIC1AREG51 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|--|
| 15 to 11 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10 | PIC1AREG5110 | Selects the signal to be input as the TSG31TSTOPC0 signal from the TSG31 timer. 0: INTENCA111 input signal of the ENCA1 timer 1: Setting prohibited. |
| 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | PIC1AREG5108 | Selects the signal to be input as the TSG31TSTOPC0 signal from TSG31 timer. 0: INTENCA011 input signal of the ENCA0 timer 1: Setting prohibited. |
| 7 | PIC1AREG5107 | Selects the signal to be input as the TS0OPC11 signal from TSG31 timer. 0: The INTTAUD117 signal input of the TAUD1 timer. 1: Setting prohibited. |
| 6, 5 | PIC1AREG51 [06:05] | Selects the signal to be input as the TSG31TSTOPS (TS1OPC10) signal from TSG31 timer. 01: The signal selected by the PIC1AREG5108 bit. 10: The signal selected by the PIC1AREG5110 bit. 11: The INTTAUD115 output of the TAUD1. Settings other than above are prohibited.*1 |
| 4 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | PIC1AREG5100 | Switches the ENCA signal and external hall sensor signal. For the note on selection of the signals, see Note 1 of Section 29.2.2.9, PIC1ATSGHALLSEL — Hall Sensor Input Select Register . 1: Select the pin input ENCA1E0, ENCA1E1, ENCA1EC. Settings other than above are prohibited.*1 |

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used

29.2.2.27 POMONSEL — Port Output Monitor Select Register

The POMONSEL register selects the channels to monitor output of TAUD0, TAUD1, TSG30, TSG31, OSTM0, and OSTM1 timers using TAUD2.

Access: This register can be read/written in 8-bit units.

Address: FFDD 7400_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|-----------|-----------|-----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | POMONSEL2 | POMONSEL1 | POMONSEL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 29.36 POMONSEL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|---|
| 7 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 to 0 | POMONSEL [2:0] | Selects the input pin to monitor. For the selectable input pins, see Table 29.37 . |

Table 29.37 Selection of Input Pins to be Monitored by the POMONSEL Register

| POMONSEL setting | 000 | 001 | 010 | 011 | 100 |
|------------------|-----------------|-------------------------|------------|-------------------------|-----------|
| ch0 | Port (TAUD2I0) | TAUD0 ch1 | TAUD0 ch0 | TAUD1 ch0* ¹ | TSG30 ch1 |
| ch1 | Port (TAUD2I1) | TAUD0 ch3 | TAUD0 ch1 | TAUD1 ch1* ¹ | TSG30 ch2 |
| ch2 | Port (TAUD2I2) | TAUD0 ch5 | TAUD0 ch2 | TAUD1 ch2* ¹ | TSG30 ch3 |
| ch3 | Port (TAUD2I3) | TAUD0 ch7 | TAUD0 ch3 | TAUD1 ch3* ¹ | TSG30 ch4 |
| ch4 | Port (TAUD2I4) | TAUD0 ch9 | TAUD0 ch4 | TAUD1 ch4* ¹ | TSG30 ch5 |
| ch5 | Port (TAUD2I5) | TAUD0 ch11 | TAUD0 ch5 | TAUD1 ch5* ¹ | TSG30 ch6 |
| ch6 | Port (TAUD2I6) | TAUD0 ch13 | TAUD0 ch6 | TAUD1 ch6* ¹ | TSG31 ch1 |
| ch7 | Port (TAUD2I7) | TAUD0 ch15 | TAUD0 ch7 | TAUD1 ch7* ¹ | TSG31 ch2 |
| ch8 | Port (TAUD2I8) | TAUD1 ch1* ¹ | TAUD0 ch8 | TAUD1 ch8 | TSG31 ch3 |
| ch9 | Port (TAUD2I9) | TAUD1 ch3* ¹ | TAUD0 ch9 | TAUD1 ch9 | TSG31 ch4 |
| ch10 | Port (TAUD2I10) | TAUD1 ch5* ¹ | TAUD0 ch10 | TAUD1 ch10 | TSG31 ch5 |
| ch11 | Port (TAUD2I11) | TAUD1 ch7* ¹ | TAUD0 ch11 | TAUD1 ch11 | TSG31 ch6 |
| ch12 | Port (TAUD2I12) | TAUD1 ch9 | TAUD0 ch12 | TAUD1 ch12 | — |
| ch13 | Port (TAUD2I13) | TAUD1 ch11 | TAUD0 ch13 | TAUD1 ch13 | — |
| ch14 | Port (TAUD2I14) | TAUD1 ch13 | TAUD0 ch14 | TAUD1 ch14 | OSTM0 |
| ch15 | Port (TAUD2I15) | TAUD1 ch15 | TAUD0 ch15 | TAUD1 ch15 | OSTM1 |

Note 1. Timer output monitor function can be used in case of the following pins is used as timer output.

P2_11 to P2_14 can be used only on 144-pin products.

TAUD1O0: P3_3, P2_11

TAUD1O1: P3_3, P2_11

TAUD1O2: P3_4, P2_12

TAUD1O3: P3_4, P2_12

TAUD1O4: P2_13

TAUD1O5: P2_13

TAUD1O6: P2_14

TAUD1O7: P2_14

29.2.2.28 PIMONSEL — Port Input Monitor Select Register

The PIMONSEL register monitors timer input.

Access: This register can be read/written in 8-bit units.

Address: FFDD 7000_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|-----------|-----------|
| | — | — | — | — | — | — | PIMONSEL1 | PIMONSEL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 29.38 PIMONSEL Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | PIMONSEL[1:0] | Selects the input pin to monitor. For the selectable input pins, see Table 29.39 . |

Table 29.39 Selection of Input Pins to be Monitored by the PIMONSEL Register

| PIMONSEL Setting | 00 | 01 | 10 |
|------------------|----------------|-----------|-----------|
| ch0 | Port (TAUJ2I0) | TAUJ0 ch0 | TAUJ1 ch0 |
| ch1 | Port (TAUJ2I1) | TAUJ0 ch1 | TAUJ1 ch1 |
| ch2 | Port (TAUJ2I2) | TAUJ0 ch2 | TAUJ1 ch2 |
| ch3 | Port (TAUJ2I3) | TAUJ0 ch3 | TAUJ1 ch3 |

29.2.2.29 SELBSSER — Synchronous Clear Enable Register

Access: This register can be read/written in 8-bit units.

Address: FFDD 2000_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---------------|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | SELBSSER[2:0] | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R/W | R/W |

Table 29.40 SELBSSER Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | |
|------------------|--------------------|--|--|---------------------------------------|------------------|------------------------------------|------------------|---|------------------|---|------------------|--|------------------|--|------------------|--------------------|
| 7 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | |
| 2 to 0 | SELBSSER[2:0] | <table border="1"> <thead> <tr> <th>SELBSSER[2:0]</th> <th>Setting for TSG3 Synchronous Clearing</th> </tr> </thead> <tbody> <tr> <td>000_B</td> <td>TSG30: Disabled TSG31: Disabled</td> </tr> <tr> <td>001_B</td> <td>TSG30: INTTAUD0114 enabled TSG31: Disabled</td> </tr> <tr> <td>010_B</td> <td>TSG30: Disabled TSG31: INTTAUD0115 enabled</td> </tr> <tr> <td>011_B</td> <td>TSG30: INTTAUD0114 enabled TSG31: INTTAUD0115 enabled</td> </tr> <tr> <td>100_B</td> <td>TSG30: INTTAUD0115 enabled TSG31: INTTAUD0115 enabled</td> </tr> <tr> <td>Other than above</td> <td>Setting prohibited</td> </tr> </tbody> </table> | SELBSSER[2:0] | Setting for TSG3 Synchronous Clearing | 000 _B | TSG30: Disabled TSG31: Disabled | 001 _B | TSG30: INTTAUD0114 enabled TSG31: Disabled | 010 _B | TSG30: Disabled TSG31: INTTAUD0115 enabled | 011 _B | TSG30: INTTAUD0114 enabled TSG31: INTTAUD0115 enabled | 100 _B | TSG30: INTTAUD0115 enabled TSG31: INTTAUD0115 enabled | Other than above | Setting prohibited |
| | | SELBSSER[2:0] | Setting for TSG3 Synchronous Clearing | | | | | | | | | | | | | |
| | | 000 _B | TSG30: Disabled TSG31: Disabled | | | | | | | | | | | | | |
| | | 001 _B | TSG30: INTTAUD0114 enabled TSG31: Disabled | | | | | | | | | | | | | |
| | | 010 _B | TSG30: Disabled TSG31: INTTAUD0115 enabled | | | | | | | | | | | | | |
| | | 011 _B | TSG30: INTTAUD0114 enabled TSG31: INTTAUD0115 enabled | | | | | | | | | | | | | |
| | | 100 _B | TSG30: INTTAUD0115 enabled TSG31: INTTAUD0115 enabled | | | | | | | | | | | | | |
| Other than above | Setting prohibited | | | | | | | | | | | | | | | |

CAUTIONS

- Using the synchronous clearing and simultaneous starting trigger functions for TSG3 at the same time is prohibited. If you are using synchronous clearing of TSG3, only use the simultaneous start trigger function at the start of counting. After counting has started, set the bits of the simultaneous start control register listed below to 0.
 - PIC1SSER0.PIC1ASSER014 (TAUD0 channel 14)
 - PIC1SSER0.PIC1ASSER015 (TAUD0 channel 15)
 - PIC1SSER2.PIC1ASSER208 (TSG30)
 - PIC1SSER2.PIC1ASSER209 (TSG31)

To start TSG30 and TSG31, and TAUD0 channels 14 and 15 simultaneously while synchronous clearing of TSG3 is in use, disable synchronous clearing of TSG3 before using the simultaneous start trigger function.

Switching of the above setting while the trigger for synchronous clearing or simultaneous starting of TSG3 is active is prohibited.
- Writing in the following ways to SSER during counter operation is prohibited.
 - Writing a value other than 000_B or the setting at the start of operation.
 - Writing at a timing that coincides with the occurrence of an interrupt for TAUD0 channel 14 or 15.

29.2.3 Function

29.2.3.1 Simultaneous Start Trigger Function

(1) Overview

This function allows any combination of timers (TAUDn, TAUJn, TSG3n, TPBA_n, OSTM_n, and ENCA_n) to be started simultaneously.

(2) Configuration

The timers which support simultaneous start trigger function are as follows.

- TAUD_n
- TAUJ_n
- TSG3_n
- TPBA_n
- OSTM_n
- ENCA_n

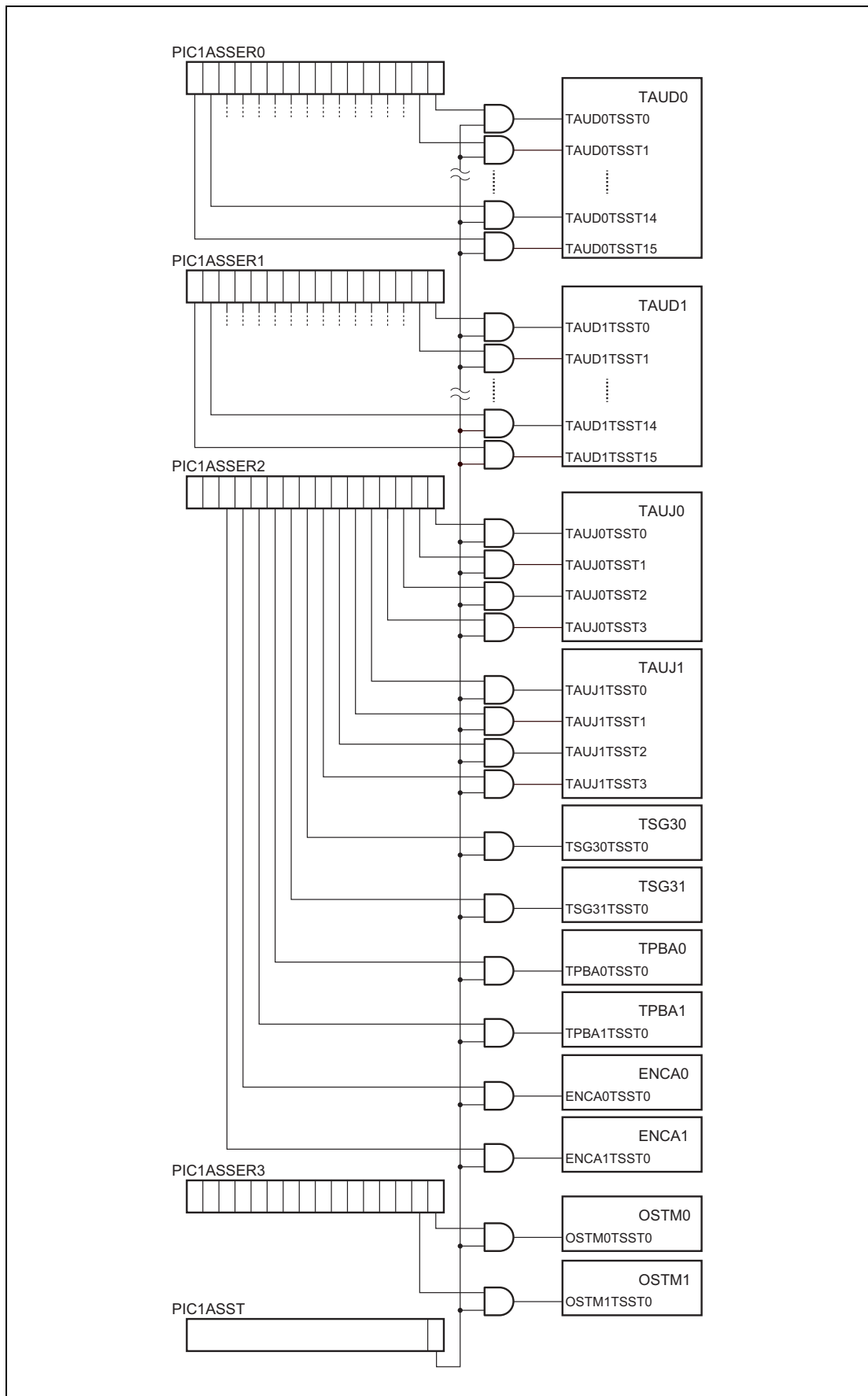


Figure 29.3 Block Diagram of Simultaneous Start Trigger Function

Set 1 to the PIC1ASYNCTRIG bit of the simultaneous start trigger control register (PIC1ASST) after unmasking the target timers. The timer operations start by active signal input to the start trigger of each timer.

(3) Registers

The PIC1A registers set by this function are as follows. For setting value of the registers, see **Section 29.2.2.2** to **Section 29.2.2.6**.

- PIC1A registers to be set
 - PIC1ASST
 - PIC1ASSER0
 - PIC1ASSER1
 - PIC1ASSER2
 - PIC1ASSER3

(4) Function

This function allows any combination of timers (TAUDn, TAUJn, TSG3n, TPBA_n, OSTM_n, and ENCA_n) to be started simultaneously.

(5) Flow Chart

The following figure shows the setting flow of this function.

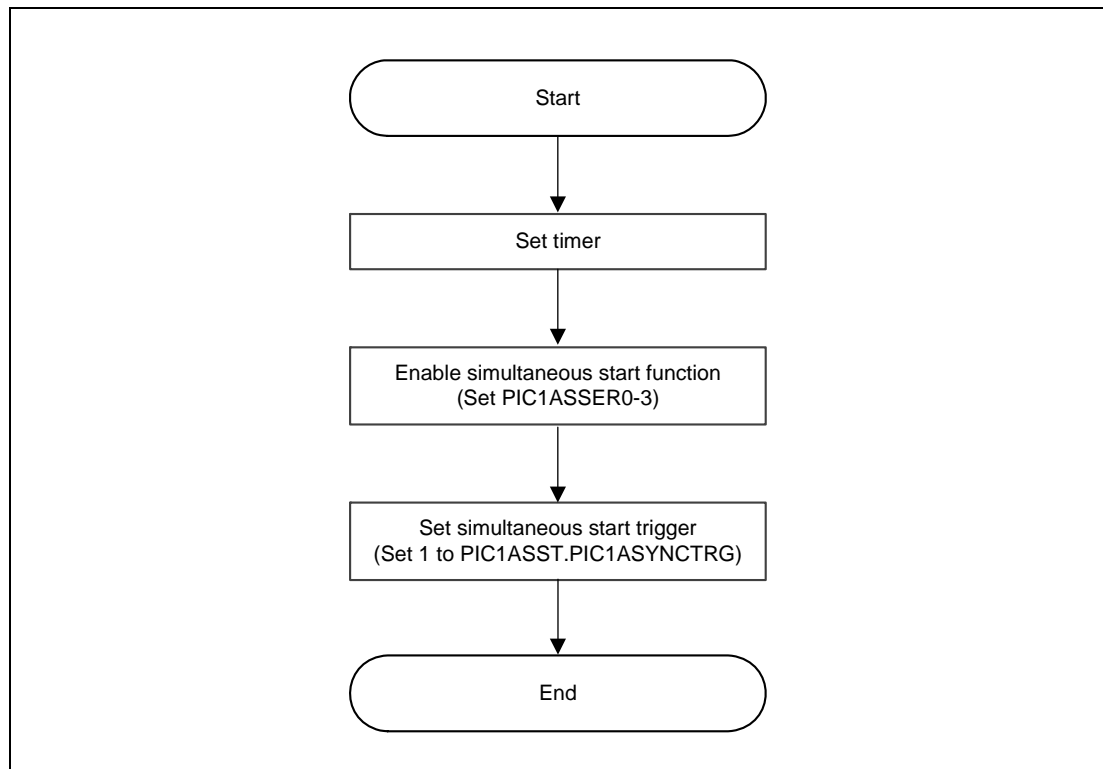


Figure 29.4 Setting Flow

Setting procedures are as follows.

- Set timer
Set the timers to start simultaneously.
- Enable simultaneous start function
Set 1 to the applicable bits of the PIC1ASSER0, PIC1ASSER1, PIC1ASSER2, and PIC1ASSER3 registers to enable simultaneous start of the corresponding timers.
- Set simultaneous start trigger
Setting 1 to the PIC1ASYNCTRG bit of the simultaneous start trigger control register (PIC1ASST) simultaneously starts the timers.

29.2.3.2 PWM Output Function with Dead Time

(1) Overview

This function generates and outputs PWM waveforms with the dead time from one phase to three phases using TAUDn.

The PWM output function of TAUD sets only the clear timing in a period under the duty ratio specification. This function enables the set timing to be specified in addition to the clear timing to output more flexible PWM waveforms with the dead time.

The following table lists the number of TAUDn channels used.

| PWM Output with Dead Time | Number of TAUDn Channels |
|--|---|
| One-phase PWM output (U phase/UB phase) | 5 channels (master channel: 1, slave channel: 4) |
| Two-phase PWM output (U phase/UB phase, V phase/VB phase) | 9 channels (master channel: 1, slave channel: 8) |
| Three-phase PWM output (U phase/UB phase, V phase/VB phase, W phase/WB phase) | 13 channels (master channel: 1, slave channel: 12) |

Note: Above are examples of PWM output combinations.

Usages of each TAUDn channel are listed in the following table. CH2 is used as the master channel.

| TAUDn Channel | U phase/ UB phase | V phase/ VB phase | W phase/ WB phase | Usage |
|---------------|----------------------|----------------------|----------------------|---------------------------------------|
| CH0 | — | — | — | Not used |
| CH1 | — | — | — | Not used |
| CH2 | √ | √ | √ | Carrier period (common to each phase) |
| CH3 | — | — | — | Not used |
| CH4 | √ | — | — | Duty (U phase setting) |
| CH5 | √ | — | — | Duty (U phase clearing) |
| CH6 | — | √ | — | Duty (V phase setting) |
| CH7 | — | √ | — | Duty (V phase clearing) |
| CH8 | — | — | √ | Duty (W phase setting) |
| CH9 | — | — | √ | Duty (W phase clearing) |
| CH10 | √ | — | — | U phase output (TOUT10) |
| CH11 | √ | — | — | UB phase output (TOUT11) |
| CH12 | — | √ | — | V phase output (TOUT12) |
| CH13 | — | √ | — | VB phase output (TOUT13) |
| CH14 | — | — | √ | W phase output (TOUT14) |
| CH15 | — | — | √ | WB phase output (TOUT15) |

Note: √: Used; —: Not used

(2) Configuration

The PWM output function with the dead time is realized by using the PWM output function/one-phase output PWM function of TAUDn and PIC1A in combination. The following figure shows the block diagram of the PWM output function with the dead time.

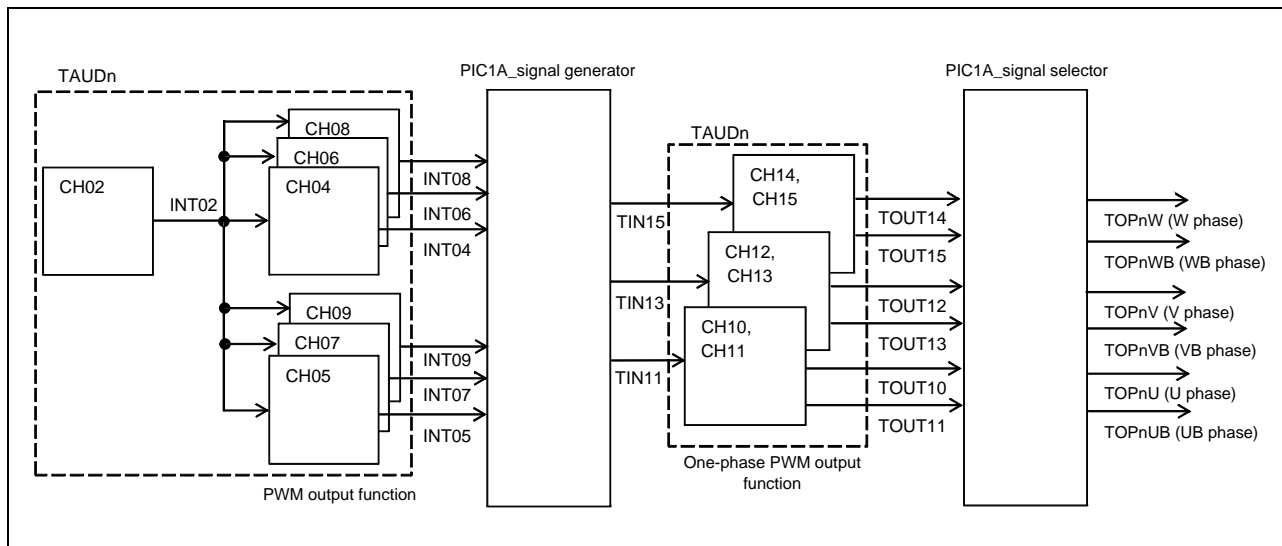


Figure 29.5 Block Diagram of PWM Output Function with Dead Time

The configuration of this function is described below using the PWM output of U phase/UB phase as an example.

- [TAUDn] PWM output function
CH02, CH04, and CH05 are used in combination. Setting the period, U phase set value, and U phase clear value to CDR02, CDR04, and CDR05, respectively, generates the set and clear PWM signals (INT04 and INT05).
- [PIC1A_signal generator] RS flip-flop circuit (RSn2)
The INT04 and INT05 inputs are selected and TIN11 (PWM signal) is generated.
- [TAUDn] One-phase PWM output function
CH10 and CH11 are used in combination. A dead time value is set to CDR11, dead time is inserted in the PWM signal to be input to TIN11, and TOUT10 (U phase PWM signal) and TOUT11 (UB phase PWM signal) are output.
- [PIC1A_signal selector]
TOUT10 and TOUT11 inputs are selected and output to TOPnU and TOPnUB pins, respectively.

A similar process occurs for V phase/VB phase and W phase/WB phase.

(3) Registers

The block diagram of PIC1A is shown in the following figure.

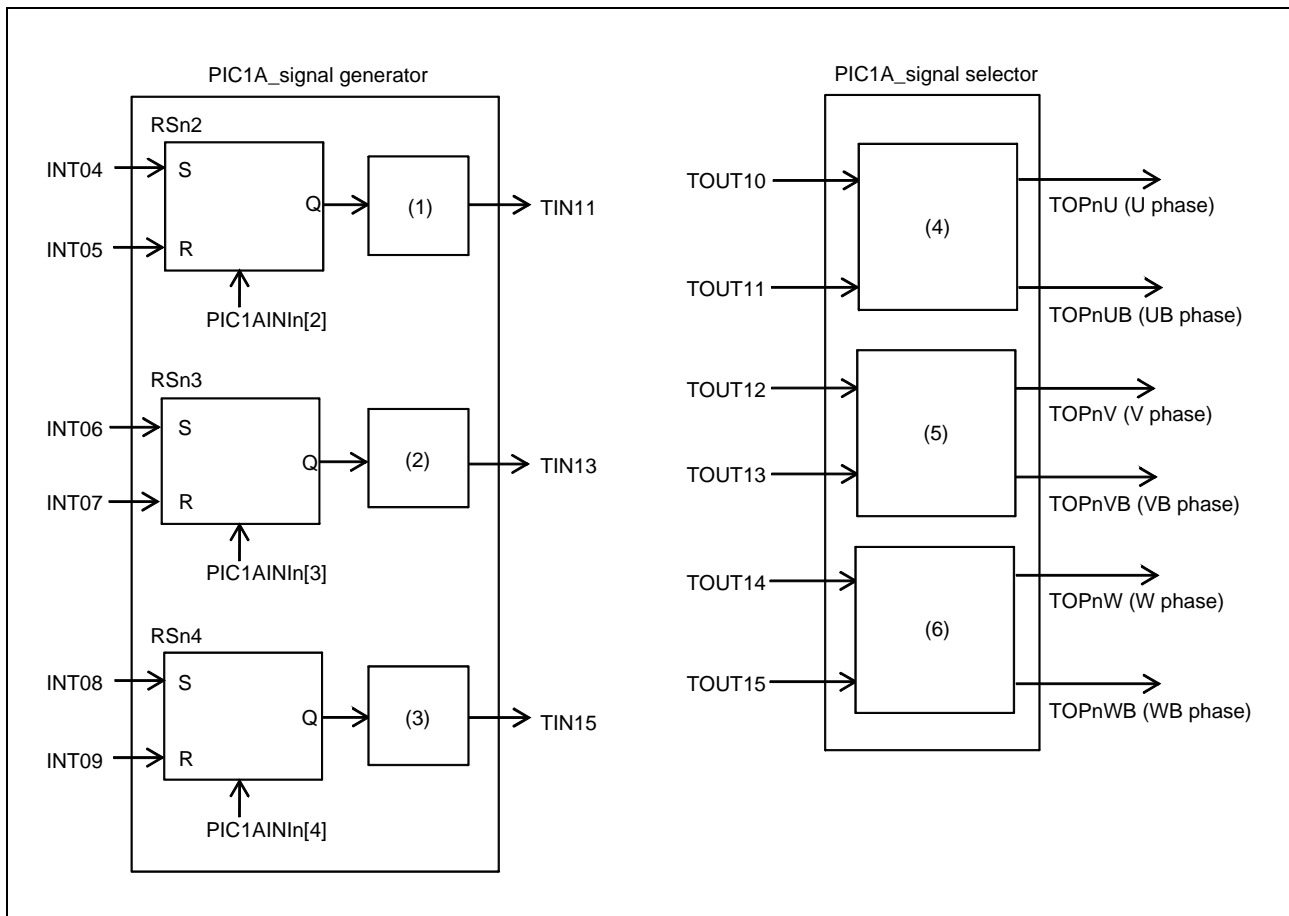


Figure 29.6 Block Diagram of PIC1A

The values of PIC1A registers used in this function are as follows.

U phase/ UB phase

The values to output the signal Q from RSn2 as TIN11. (Figure 29.6, unit (1))

$$\text{PIC1AREG2n2}[19:18] = 10_{\text{B}}$$

$$\text{PIC1AREG2n2}[2] = 1_{\text{B}}$$

$$\text{PIC1ATAUDnSEL}[23:22] = 00_{\text{B}}$$

The values to output TOUT10 and TOUT11 as TOPnU and TOPnUB, respectively (Figure 29.6, unit (4))

$$\text{PIC1AREG2n1}[19:16] = 0000_{\text{B}}$$

$$\text{PIC1AREG2n3}[2:0] = 000_{\text{B}}$$

$$\text{PIC1AREG2n3}[6:4] = 000_{\text{B}}$$

V phase/ VB phase

The values to output the signal Q from RSn3 as TIN13. (**Figure 29.6**, unit (2))

$$\text{PIC1AREG2n2}[23:22] = 10_{\text{B}}$$

$$\text{PIC1AREG2n2}[3] = 1_{\text{B}}$$

$$\text{PIC1ATAUDnSEL}[27:26] = 00_{\text{B}}$$

The values to output TOUT12 and TOUT13 as TOPnV and TOPnVB, respectively (**Figure 29.6**, unit (5))

$$\text{PIC1AREG2n1}[23:20] = 0000_{\text{B}}$$

$$\text{PIC1AREG2n3}[10:8] = 000_{\text{B}}$$

$$\text{PIC1AREG2n3}[14:12] = 000_{\text{B}}$$

W phase/ WB phase

The values to output the signal Q from RSn4 as TIN15. (**Figure 29.6**, unit (3))

$$\text{PIC1AREG2n2}[27:26] = 10_{\text{B}}$$

$$\text{PIC1AREG2n2}[4] = 1_{\text{B}}$$

$$\text{PIC1ATAUDnSEL}[31:30] = 00_{\text{B}}$$

The values to output TOUT14 and TOUT15 as TOPnW and TOPnWB, respectively (**Figure 29.6**, unit (6))

$$\text{PIC1AREG2n1}[27:24] = 0000_{\text{B}}$$

$$\text{PIC1AREG2n3}[18:16] = 000_{\text{B}}$$

$$\text{PIC1AREG2n3}[22:20] = 000_{\text{B}}$$

Enables initialization of RSn2 to RSn4

The values to enable initialization of RSn2 to RSn4

$$\text{PIC1AINIn0}[4] = 1_{\text{B}} \text{ (initialized)}$$

$$\text{PIC1AINIn0}[3] = 1_{\text{B}} \text{ (initialized)}$$

$$\text{PIC1AINIn0}[2] = 1_{\text{B}} \text{ (initialized)}$$

(4) Function

Details of this function are described using the one-phase PWM output (U phase/UB phase) with dead time as an example.

The following figure shows the timing diagram.

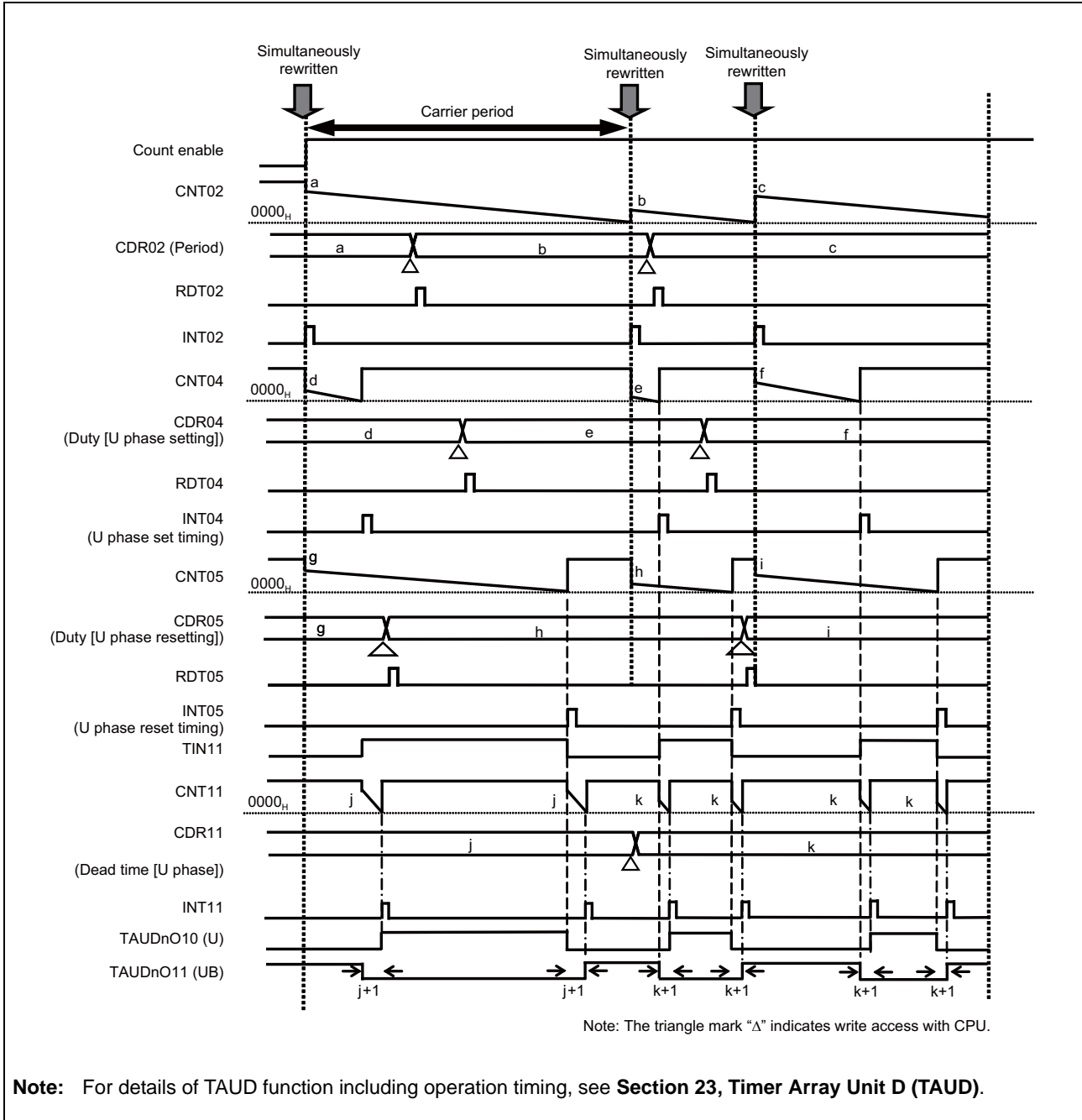


Figure 29.7 Timing Diagram of One-Phase PWM Output with Dead Time (U phase/ UB phase)

- (1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.*¹
- (2) For CH04 and CH05, the values set to CDR04 and CDR05 are reloaded to CNT04 and CNT05, respectively, upon the occurrence of a CH02 underflow.
- (3) INT04 is generated and TIN11 becomes high level upon the occurrence of a CH04 underflow, and INT05 is generated and TIN11 becomes low level upon the occurrence of a CH05 underflow. This process generates a PWM waveform.
- (4) The set values are reloaded to CNT11 at both edges of TIN11.*²
- (5) INT11 is generated and TAUDnO10 becomes high level upon the occurrence of a CH11 underflow, and INT05 is generated and TAUDnO10 becomes low level upon the occurrence of a CH05 underflow. This process generates a U-phase PWM waveform, which is output to TOPnU.
- (6) TAUDnO11 becomes low level at the rising edge of TIN11, and INT11 is generated and TAUDnO11 becomes high level upon the occurrence of a CH11 underflow. This process generates a UB-phase PWM waveform, which is output to TOPnUB.

A similar process occurs for V phase/VB phase and W phase/WB phase.

Note 1. Select the count clock signal of the same clock for TAUDn.

Note 2. Set both edges (rising and falling) as the valid edge of TIN11 of TAUDn to be detected.

Details of this function when setting a clear timing longer than the carrier period are described below using V phase/VB phase as an example.

The following figure shows the timing diagram.

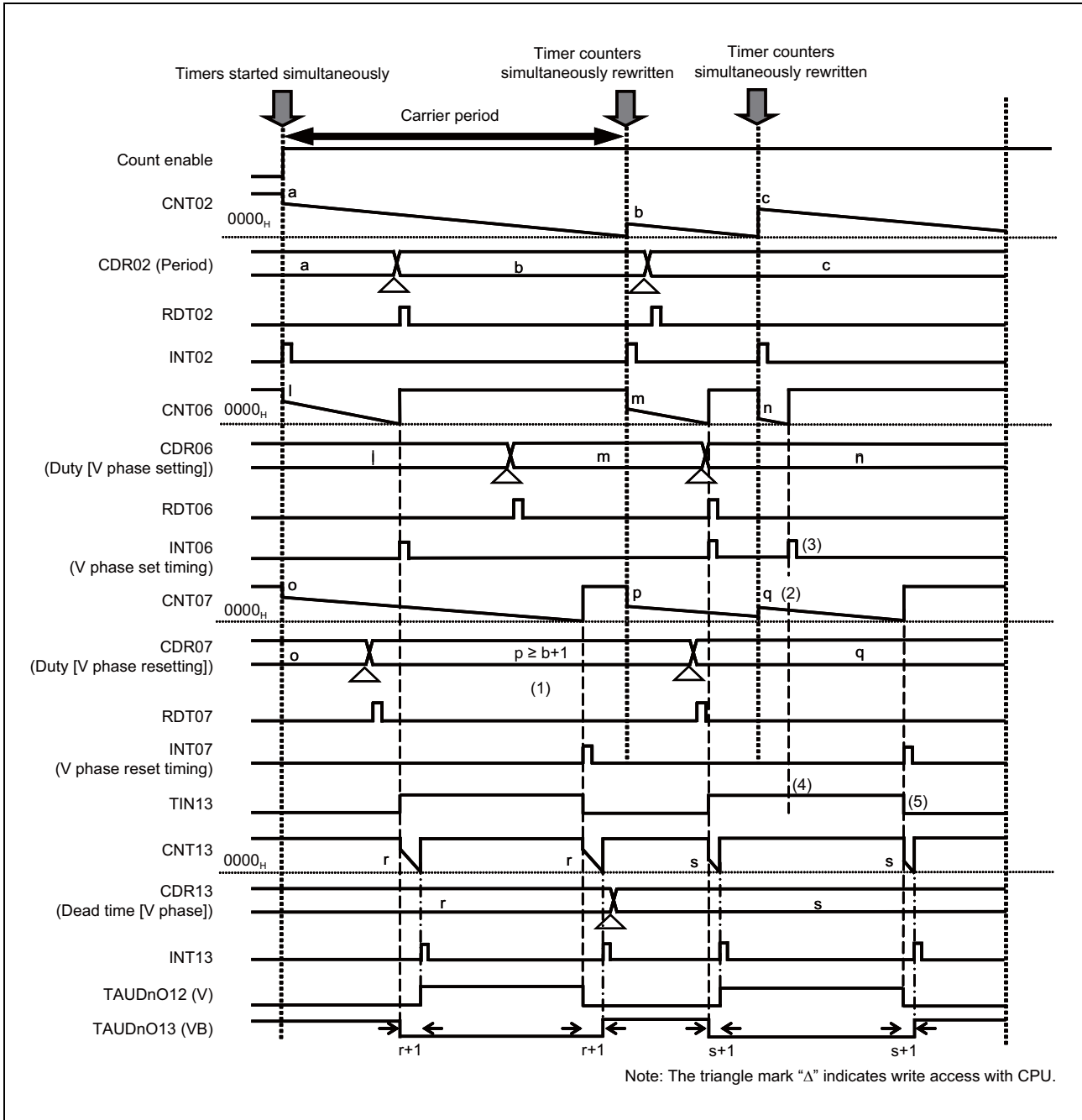


Figure 29.8 Clear Timing Value > Carrier Period Value (V phase/VB phase)

Setting a clear timing longer than the carrier period allows the waveform output to be extended over the carrier period.

An operation example of one-phase PWM output (V phase/VB phase) is shown below. The operation flow from timer operation start to one-phase PWM output by one-phase PWM output function, refer to the description for one-phase PWM output with dead time (U phase/UB phase).

When the value set in CH07 is larger than the value set in CH02 (**Figure 29.8 (1)**), underflow of the carrier period timer is generated before generation of V phase clear timing signal (INT07), and the value is reloaded (**Figure 29.8 (2)**).

As a result, the V phase clear timing signal (INT07) that should be generated is not generated, and instead the V phase set timing signal (INT06) is generated again (**Figure 29.8 (3)**).

At this time, the PWM waveform is not affected because V phase set timing signal is ignored in the PIC circuit (**Figure 29.8 (4)**). Therefore, the PWM waveform is output extended over the carrier period (**Figure 29.8 (5)**)

The following figure shows the timing diagram of three-phase PWM output with the dead time.

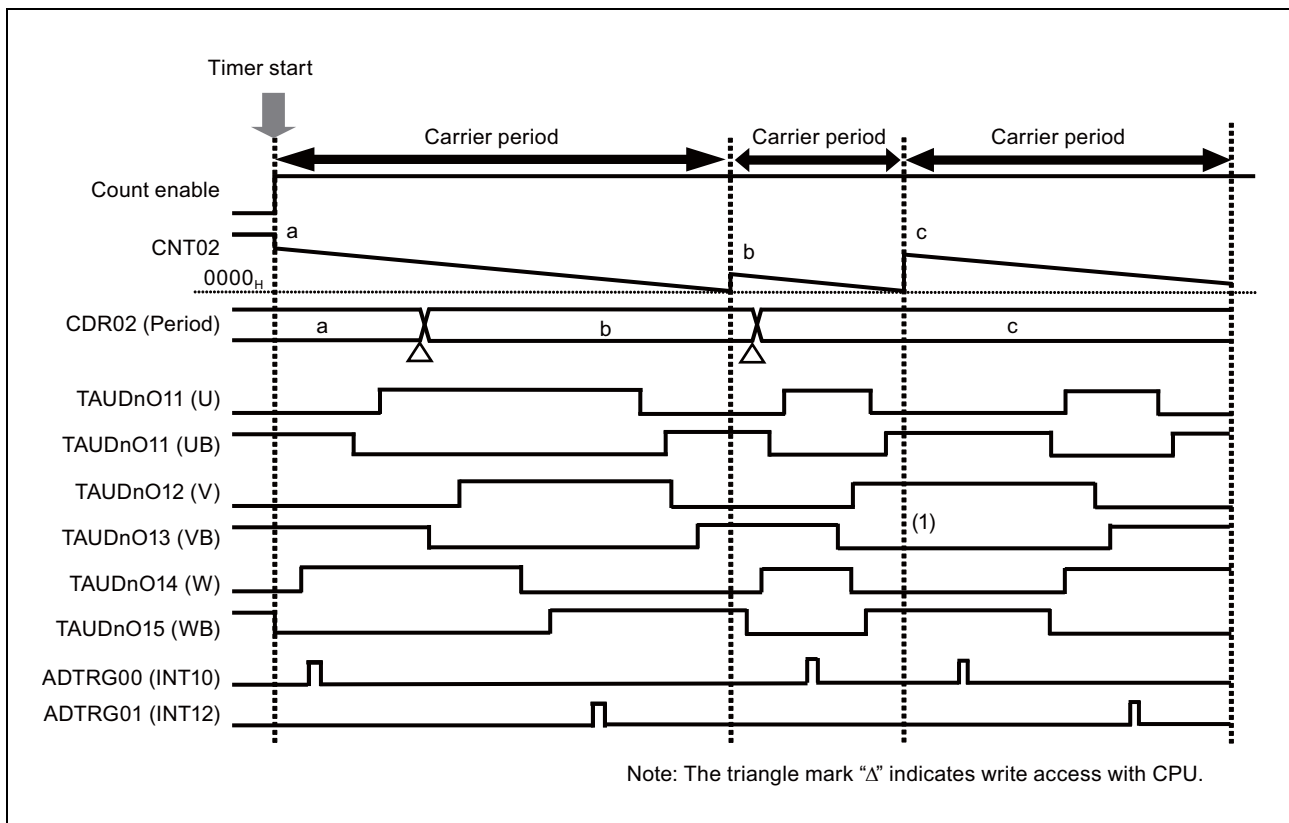
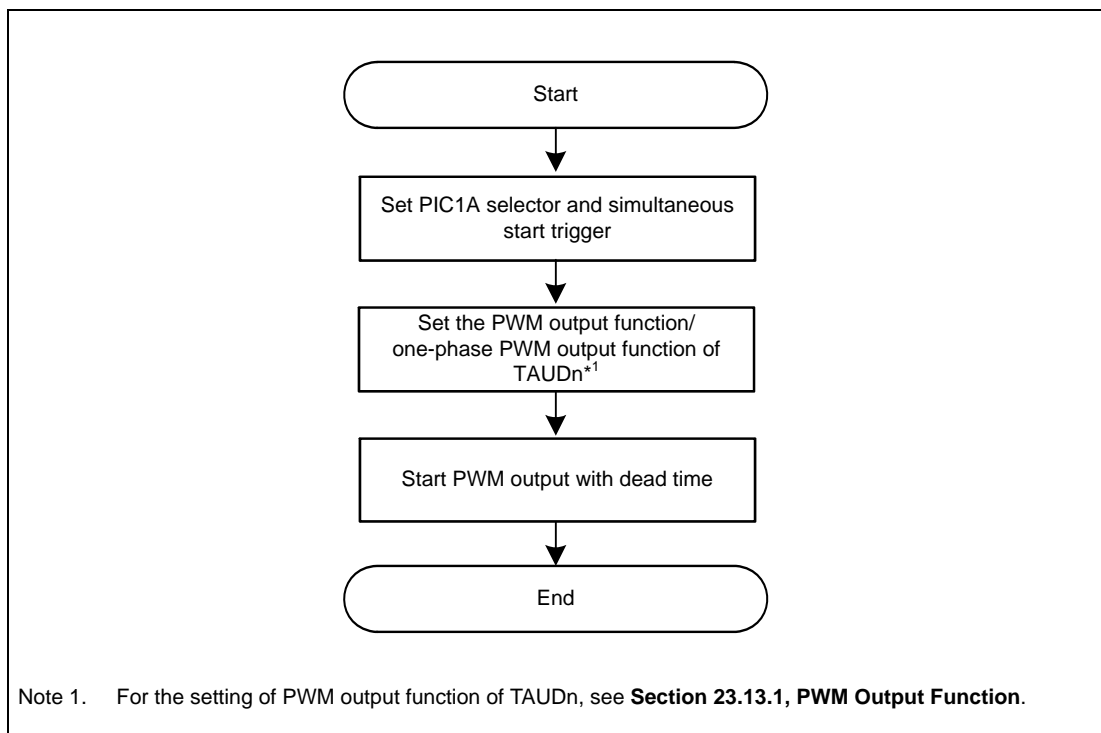


Figure 29.9 Timing Diagram of Three-Phase PWM Output with Dead Time

(5) Flow Chart

The following flow chart shows the PMW output function with the dead time.



29.2.3.3 High Accuracy Triangle Wave PWM Output Function with Dead Time

(1) Overview

This function generates triangle wave PWM output with dead time from one-phase to three-phases. Compared to the triangle wave PWM output function with dead time of TAUDn, this function enables a control of the variable dead time range, where duty cycle is close to 100% and 0%.

The TAUDn channels used in this function and the numbers of channels are listed in the following table.

| High Accuracy Triangle Wave PWM Output with Dead Time | Number of TAUDn Channels |
|--|---|
| One-phase PWM output (U phase/UB phase) | 5 channels (master channel: 1, slave channel: 4) |
| Two-phase PWM output (U phase/UB phase, V phase/VB phase) | 9 channels (master channel: 1, slave channel: 8) |
| Three-phase PWM output (U phase/UB phase, V phase/VB phase, W phase/WB phase) | 13 channels (master channel: 1, slave channel: 12) |

Note: Above are examples of PWM output combinations.

Usages of each TAUDn channel are listed in the following table. CH2 is used as the master channel of CH3 to 9.

CHm is used as the master channel of CHm+1 (m = 10, 12, 14).

| TAUDn Channel | U phase / UB phase | V phase / VB phase | W phase / WB phase | Usage |
|---------------|--------------------|--------------------|--------------------|--|
| CH0 | — | — | — | Not used |
| CH1 | — | — | — | Not used |
| CH2 | √ | √ | √ | Carrier period (common to each phase) |
| CH3 | — | — | — | Not used |
| CH4 | √ | — | — | Triangle PWM output with dead time (U phase/UB phase) |
| CH5 | √ | — | — | |
| CH6 | — | √ | — | Triangle PWM output with dead time (V phase/VB phase) |
| CH7 | — | √ | — | |
| CH8 | — | — | √ | Triangle PWM output with dead time (W phase/WB phase) |
| CH9 | — | — | √ | |
| CH10 | √ | — | — | Reduced dead time pulse (U phase/UB phase) |
| CH11 | √ | — | — | |
| CH12 | — | √ | — | Reduced dead time pulse (V phase/VB phase) |
| CH13 | — | √ | — | |
| CH14 | — | — | √ | Reduced dead time pulse (W phase/WB phase) |
| CH15 | — | — | √ | |

Note: √: Used; —: Not used

(2) Configuration

The high accuracy triangle wave PWM output function with the dead time is realized by using the triangle wave PWM output function/one-shot pulse output function of TAUDn and PIC1A in combination. The following figure shows the block diagram of the high accuracy triangle wave PWM output function with the dead time.

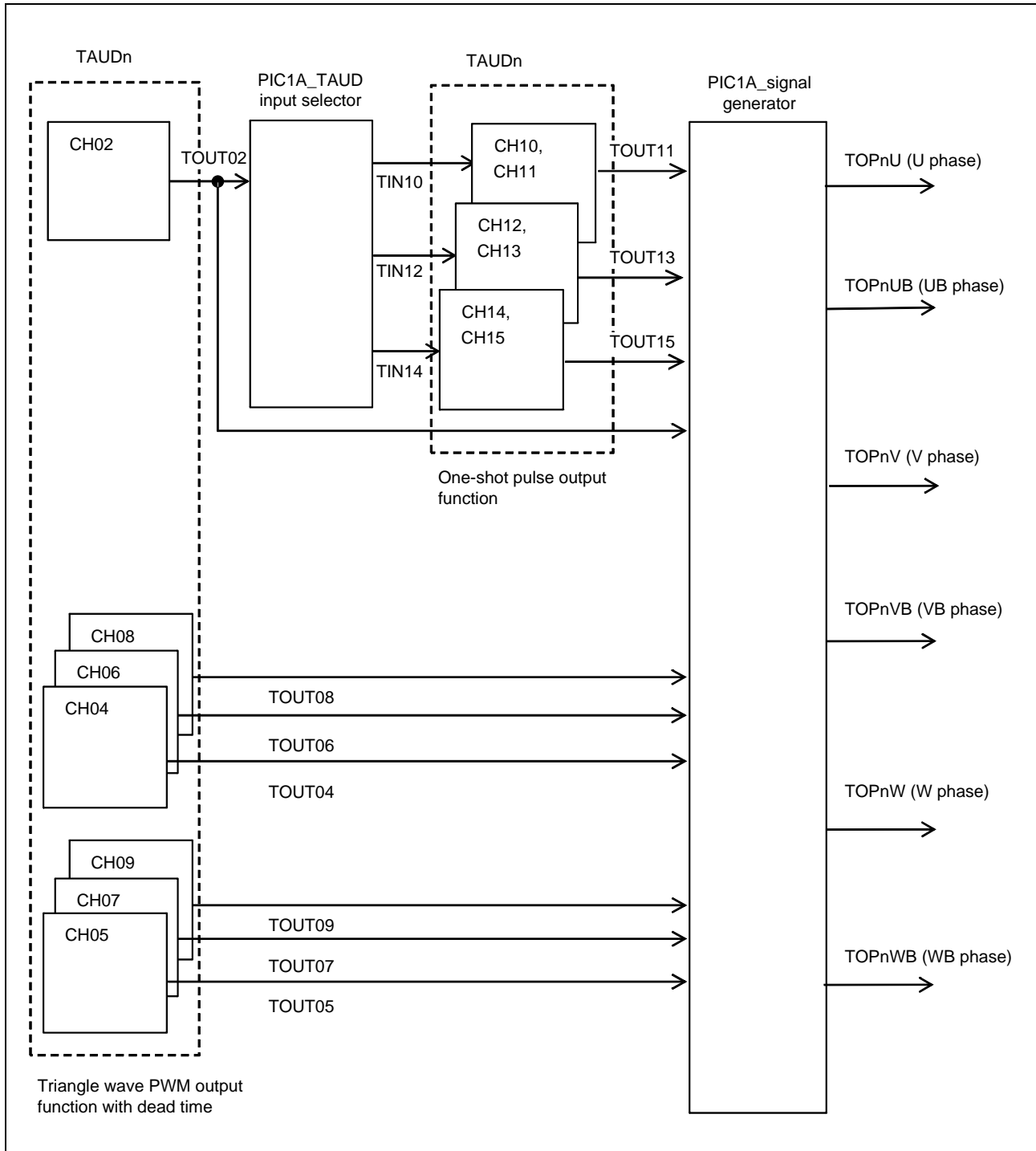


Figure 29.10 Block Diagram of High Accuracy Triangle Wave Three-Phase PWM Output With Dead Time

The configuration of this function is described below using the PWM output of U phase/UB phase as an example.

- [PIC1A_TAUD input selector]
TOUT02 is selected and output to TIN10.
- [TAUDn] One-shot pulse output function
CH10 and CH11 are used in combination. The delay value and the pulse width are set to CDR10 and CDR11, respectively, and the one-shot pulse output signal (TOUT11) is generated.
- [TAUDn] Triangle wave PWM output function with dead time
CH02, CH04, and CH05 are used in combination. The period, duty, and dead time are set to CDR02, CDR04, and CDR05, respectively, and a triangle wave PWM output signal with dead time (TOUT04 and TOUT05) is generated.
- [PIC1A_signal generator]
The reduced dead time pulses (UO1 and UO2) are generated at PFN001 from the one-shot pulse output signal.
UO1 and UO2 are synthesized with TOUT04 and TOUT05 at FN00 and FN01, respectively, a dead time variable range pulse is added, and TOPnU (U phase PWM signal) and TOPnUB (UB phase PWM signal) are generated.

A similar process occurs for V phase/VB phase and W phase/WB phase.

(3) Registers

The following figure shows the block diagram of PIC1A.

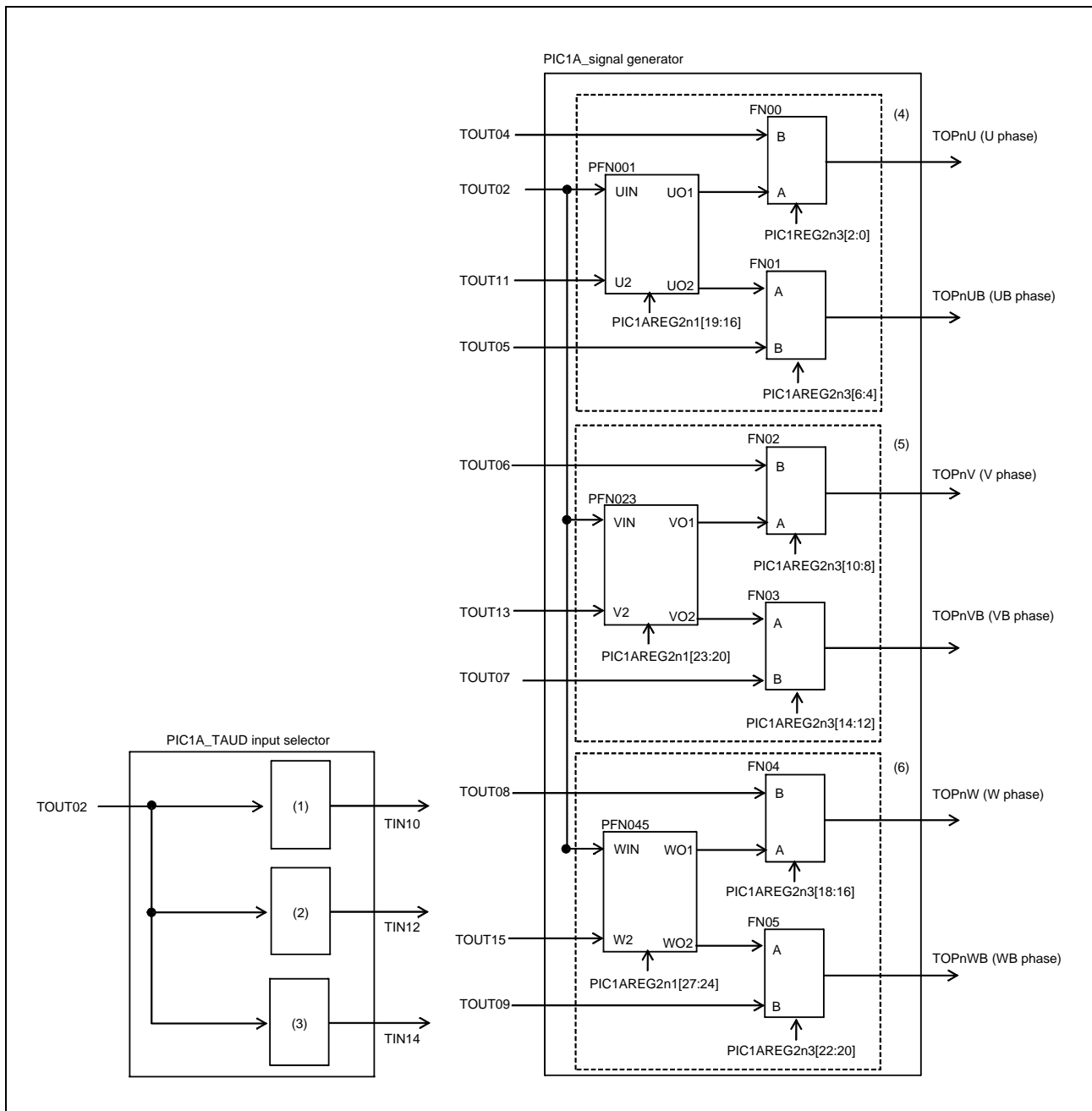


Figure 29.11 Block Diagram of PIC1A

The values of PIC1A registers used in this function are as follows.

- (1) PIC1A_TAUD input selector (U phase/ UB phase)
 The values to output TOUT02 as TIN10 (common to active high/low)
 $PIC1AREG2n0[18] = 1_B$
 $PIC1AREG2n2[17:16] = 10_B$
 $PIC1ATAUDnSEL[21:20] = 00_B$
- (2) PIC1A_TAUD input selector (V phase/ VB phase)
 The values to output TOUT02 as TIN12 (common to active high/low)
 $PIC1AREG2n0[18] = 1_B$
 $PIC1AREG2n2[21:20] = 10_B$
 $PIC1ATAUDnSEL[25:24] = 00_B$
- (3) PIC1A_TAUD input selector (W phase/ WB phase)
 The values to output TOUT02 as TIN14 (common to active high/low)
 $PIC1AREG2n0[18] = 1_B$
 $PIC1AREG2n2[25:24] = 10_B$
 $PIC1ATAUDnSEL[29:28] = 00_B$
- (4) PIC1A_ signal generator (U phase/ UB phase)
 The values to output one-phase PWM (active high/low) from TAUDnO10 and TAUDnO11
 $PIC1AREG2n1[19:16] = 1010_B$ (active high), 1111_B (active low)
 $PIC1AREG2n3 [06:04] = 100_B$ (active high), 101_B (active low)
 $PIC1AREG2n3[02:00] = 100_B$ (active high), 101_B (active low)
- (5) PIC1A_ signal generator (V phase/ VB phase)
 The values to output one-phase PWM (active high/low) from TAUDnO12 and TAUDnO13
 $PIC1AREG2n1[23:20] = 1010_B$ (active high), 1111_B (active low)
 $PIC1AREG2n3 [14:12] = 100_B$ (active high), 101_B (active low)
 $PIC1AREG2n3[10:08] = 100_B$ (active high), 101_B (active low)
- (6) PIC1A_ signal generator (W phase/ WB phase)
 The values to output one-phase PWM (active high/low) from TAUDnO14 and TAUDnO15
 $PIC1AREG2n1[27:24] = 1010_B$ (active high), 1111_B (active low)
 $PIC1AREG2n3 [22:20] = 100_B$ (active high), 101_B (active low)
 $PIC1AREG2n3[18:16] = 100_B$ (active high), 101_B (active low)

(4) Function

Details of this function are described using U phase/UB phase as an example. The function with V phase/VB phase and W phase/WB phase are operated under the same logic as that of U phase/UB phase with different input signals and register settings.

- U phase combination circuit (PFN001)
This circuit generates reduced dead time pulses*¹ (FN00A and FN01A) that are used to insert the pulse generated by the one-shot pulse output function into the triangle wave PWM generated by the triangle wave PWM output function with the dead time. For the block diagram, see **Figure 29.1, Block Diagram of PFN001**.

Note 1. Reduced dead time pulses are the pseudo pulses that are inserted into the PWM output provided by the triangle wave PWM output function with dead time of TAUDn and that are modeled on the dead time pulses that are generated in the range where duty cycle is close to 100% or 0% in PWM output in HT-PWM mode of TSG3n.

- Logical operation circuits (FN0i) (i = 0, 1)
This circuit synthesizes the triangle wave PWM output (TOUT04 and TOUT05) from the triangle wave PWM output function with the dead time and the outputs from the combinational circuit PFN001 (UO0 and UO1) to generate PWM with variable dead time range pulse inserted. The synthesizing logic of this circuit is selected with PIC1AREG2n3k (k = 00 to 02, 04 to 06). For the block diagram, see **Figure 29.2, Block Diagram of FN00**.

Details of this function are described using high accuracy triangle wave PWM output function (U phase/UB phase) as an example.

The following figure shows the timing diagram when U phase duty cycle = 0% and UB phase duty cycle = 100% at active high.

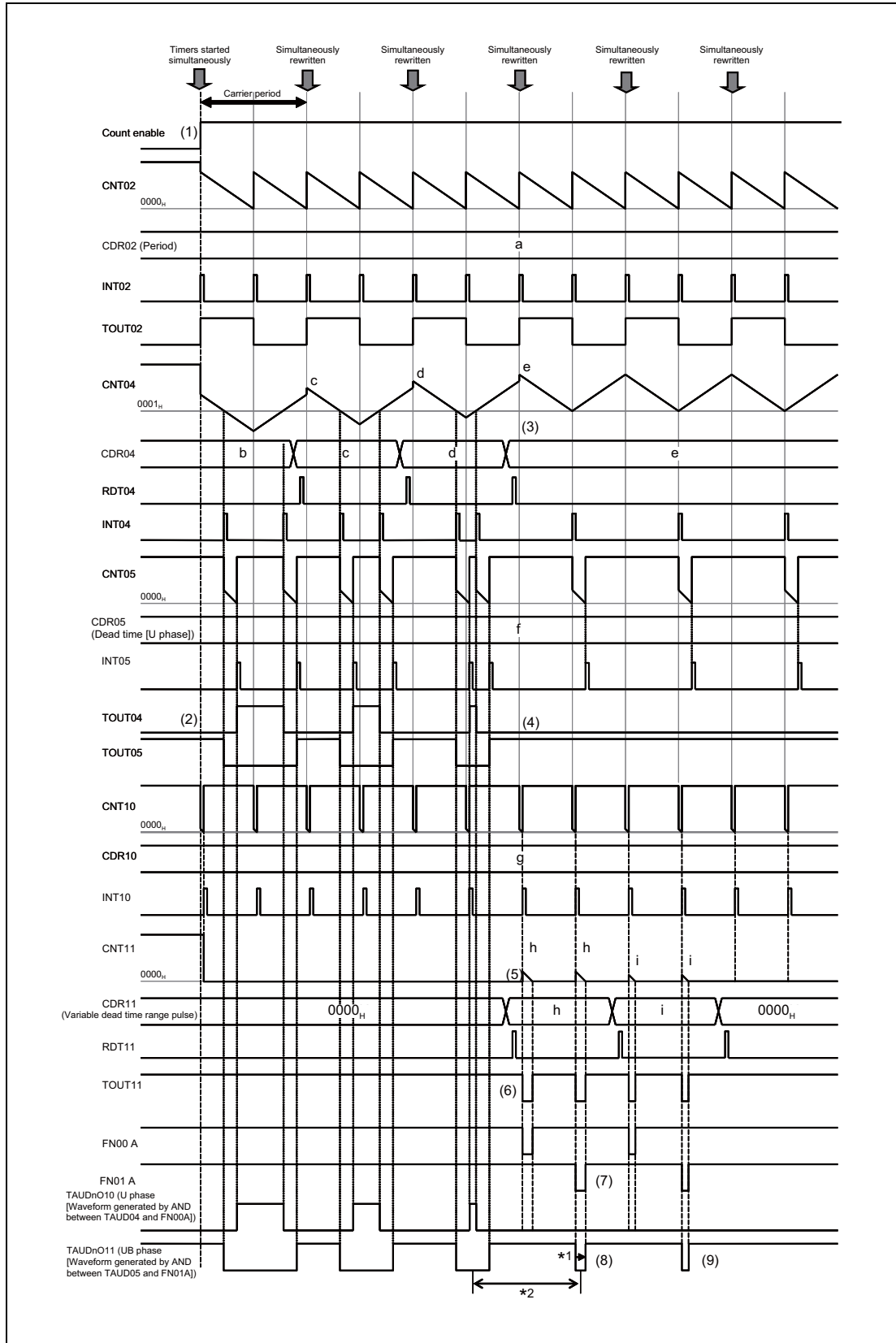


Figure 29.12 Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 0%, UB phase duty cycle = 100%) Output with Dead Time (Active High)

- (1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.
- (2) TOUT04 and TOUT05 are generated by the triangle wave PWM output function with dead time.
- (3) A 0% duty cycle value is set in CDR04 for U phase output.
- (4) The setting described in step (3) sets the TOUT04 output to the inactive level and the TOUT05 output to the active level.
- (5) To generate the variable dead time range pulse, a value for the reduced dead time pulse width is set in CDR11 when a 0% duty cycle value for U phase output is set in step (3).
- (6) CH10 count starts with the effective edge of TOUT02 and INT10 is generated when the counter underflows. CH11 count starts by generation of INT10 and the reduced dead time pulse (TOUT11) of the width set in CDR11 is output.
- (7) The reduced dead time pulse (UO1 and UO2) are generated from TOUT02 and TOUT11 in PFN001.
- (8) UO1 and UO2 are synthesized with TOUT04 and TOUT05 in FN00 and FN01, and are output as TOPnU (U phase PWM signal) and TOPnUB (UB phase PWM signal).

CAUTION

Since the reduced dead time pulses are sawtooth waves, they expand and contract on one side, unlike the triangle wave pulses, which expand and contract on both sides. Since one-side expansion and contraction applies to the reduced dead time pulses, a one-phase PWM output period in the variable dead time range is longer by half the width of the inserted reduced dead time pulse.

The following figure shows the timing diagram when U phase duty cycle = 100% and UB phase duty cycle = 0% at active high.

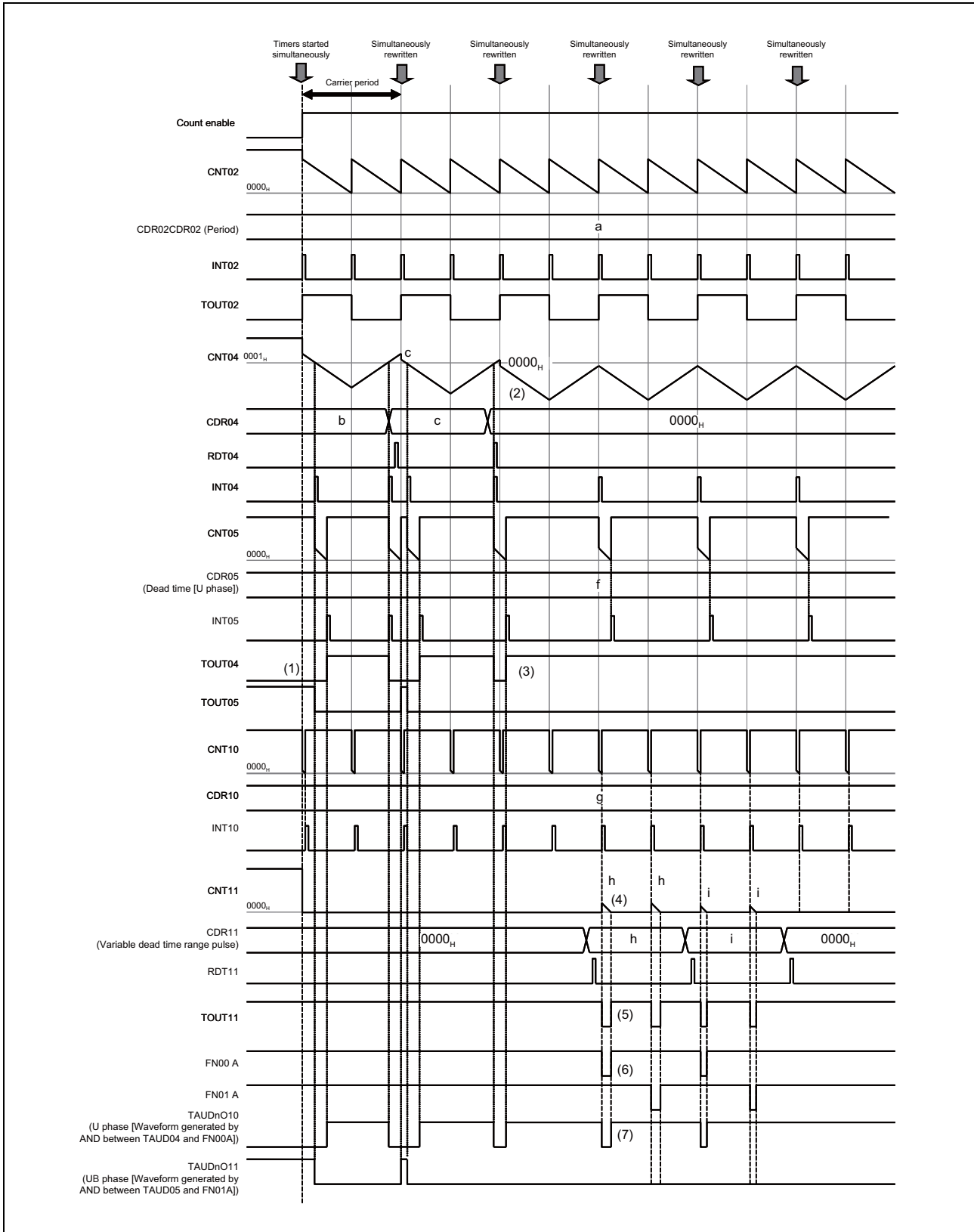


Figure 29.13 Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 100%, UB phase duty cycle = 0%) Output with Dead Time (Active High)

- (1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.
- (2) TOUT04 and TOUT05 are generated by the triangle wave PWM output function with dead time.
- (3) A 100% duty cycle value ($CDR4 = 0000_H$) is set in CDR04 for U phase output.
- (4) TOUT04 outputs the active level and TOUT05 outputs the inactive level.
- (5) A value for the reduced dead time pulse width is set in CDR11 one period after setting 100% duty cycle for U phase output.
- (6) CH10 count starts with the effective edge of TOUT02 and INT10 is generated when the counter underflows. CH11 count starts by generation of INT10 and the reduced dead time pulse (TOUT11) of the width set in CDR11 is output.
- (7) The reduced dead time pulse (UO1 and UO2) are generated from TOUT02 and TOUT11 in PFN001.
- (8) UO1 and UO2 are synthesized with TOUT04 and TOUT05 in FN00 and FN01, and are output as TOPnU (U phase PWM signal) and TOPnUB (UB phase PWM signal).

CAUTION

As shown in Figure 29.14, if a value is set for the variable dead time range pulse width in CDR11 at the same time as a 100% duty cycle value is set for U phase output in CDR04, the variable dead time range pulse affects the last PWM output from TOUT04 (indicated by Figure 29.14, (1)) by the amount of time indicated by Figure 29.14, (2). CDR11 must be set one period after setting CDR04.

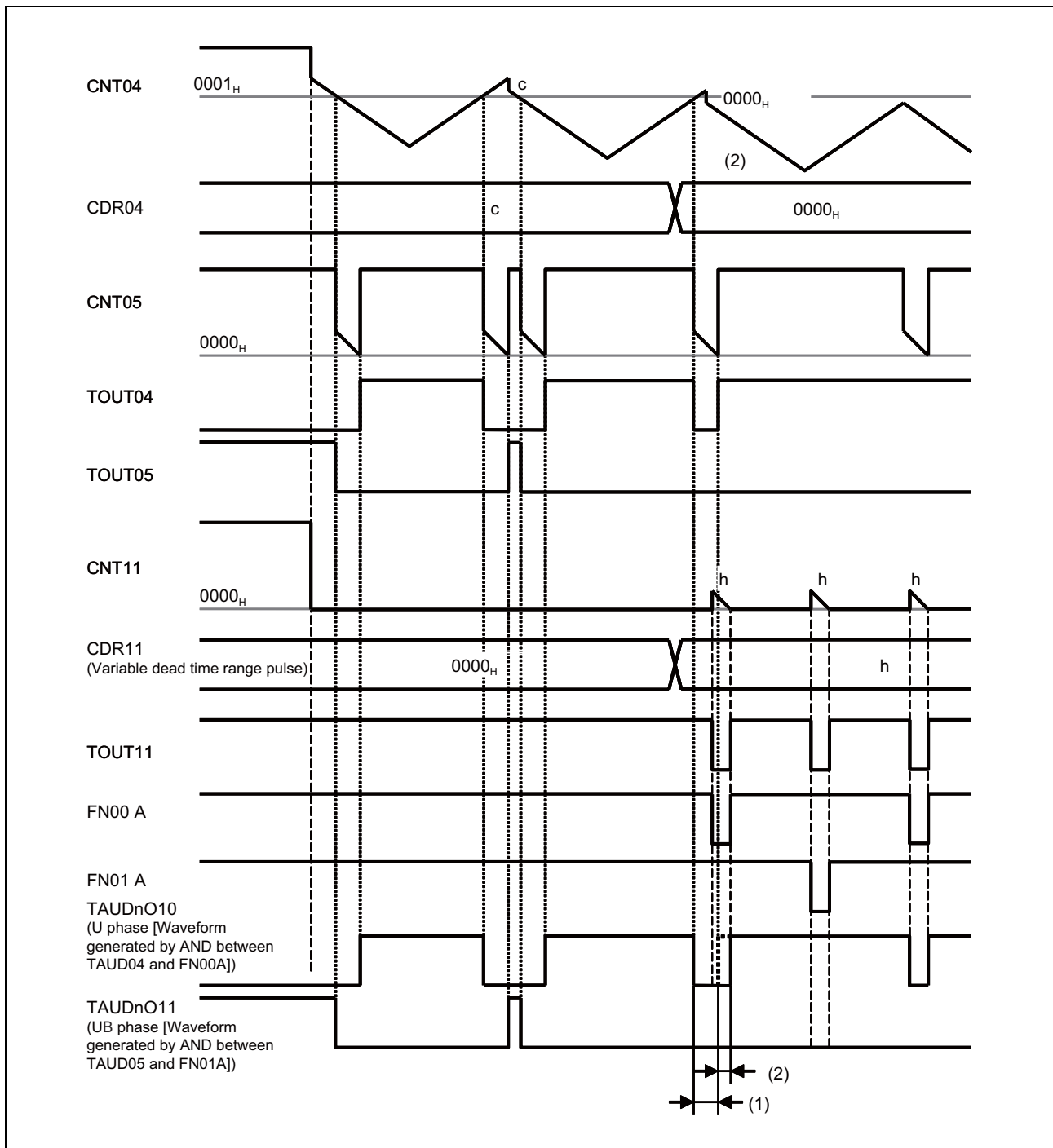


Figure 29.14 Timing Diagram of Variable Dead Time Range Pulse Affecting Triangle Wave PWM Output with Dead Time

The following figure shows the timing diagram when U phase duty cycle = 100% and UB phase duty cycle = 0% at active low.

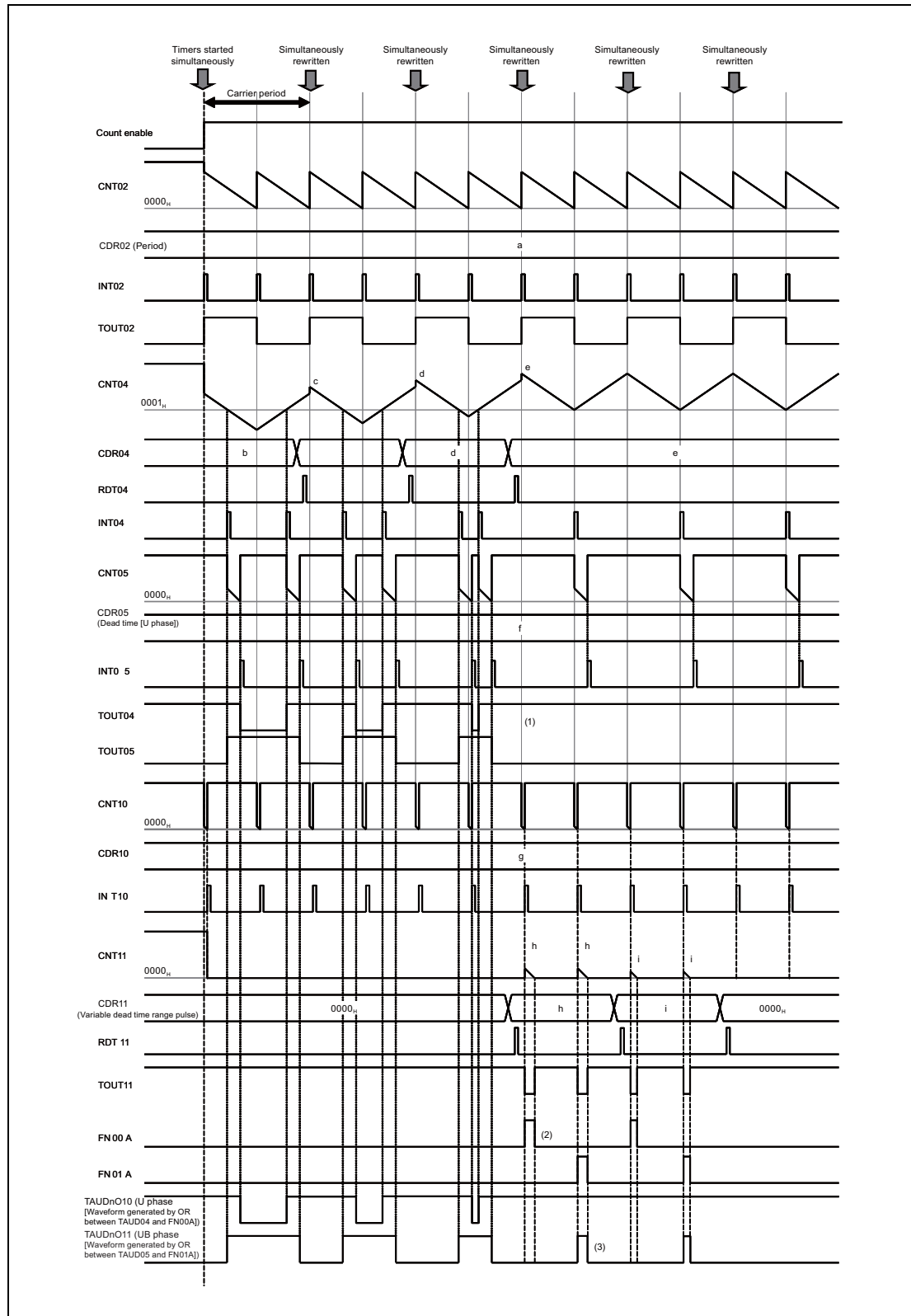


Figure 29.15 Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 100%, UB phase duty cycle = 0%) Output with Dead Time (Active Low)

The operation flow from timer operation start to triangle wave PWM output with dead time is same as **Figure 29.13, Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 100%, UB phase duty cycle = 0%) Output with Dead Time (Active High)**, with the difference of the PWM output signal from TOUT04 and TOUT05 are active low.

CAUTION

Set each CDR for the one-shot pulse output function so that the following condition is satisfied.

$$\text{CDR05} \geq (\text{CDR10} + \text{CDR11})$$

If the condition above is not satisfied, the output waveform may be influenced. To minimize the influence, satisfy the condition above, and also fix the value of CDR11 to 0000_H until the reduced dead time pulse is required.

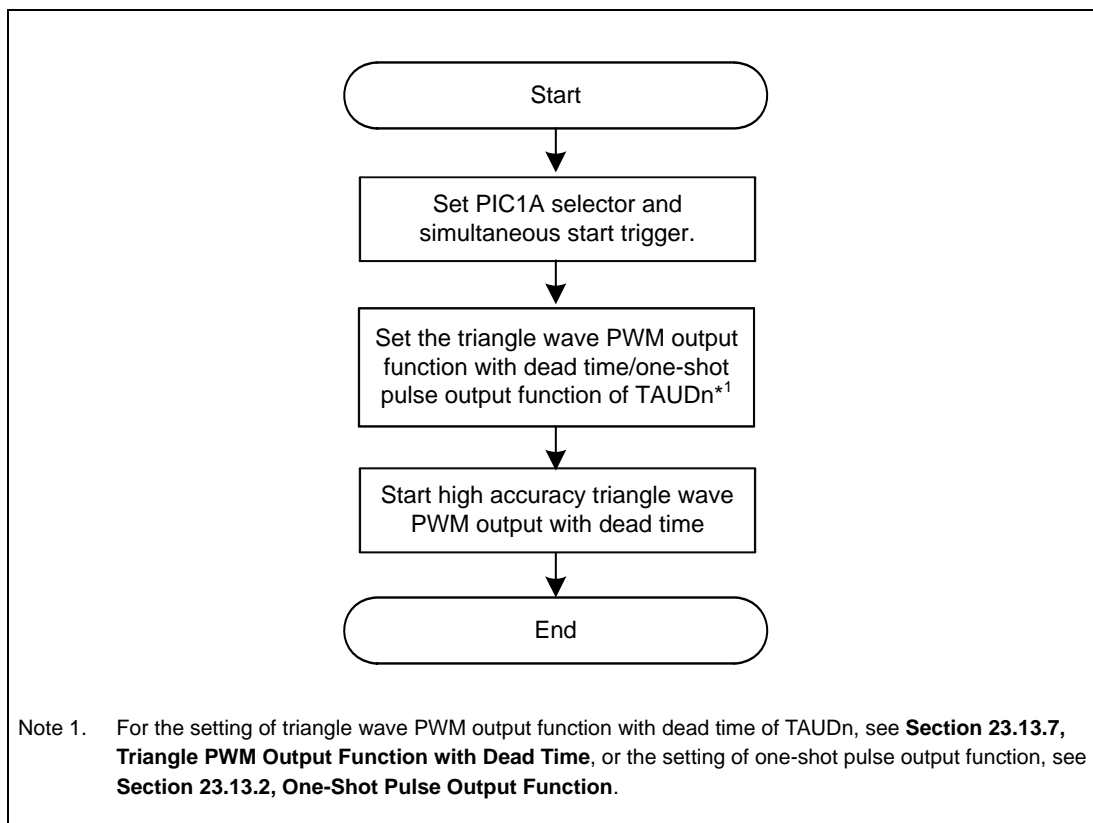
Set the both edges to be detected of TIN10 (TOUT02) as effective, and set TAUDnTOL11 to 1 (active low).

Select the count clock signal (CK0 to 3) of the same clock for TAUDn.

After high accuracy triangle wave PWM output with dead time is started, do not set the value of variable dead time pulse width at the same time as setting a 100% duty cycle value for U phase, V phase, and W phase.

(5) Flow Chart

The flow chart of this function is shown in the following figure.



29.2.3.4 Delay Pulse Output Function with Dead Time

(1) Overview

This function allows generation of PWM output with the dead time, that is, delayed as specified from the period timing using TAUDn.

With this function, PWM output can be reset in the next period unlike with the function described in **Section 29.2.3.2, PWM Output Function with Dead Time**.

The following table lists the number of channels used.

| PWM Output with Dead Time | Number of TAUDn Channels |
|--|---|
| One-phase PWM output (U phase/UB phase) | 5 channels (master channel: 1, slave channel: 4) |
| Two-phase PWM output (U phase/UB phase, V phase/VB phase) | 9 channels (master channel: 1, slave channel: 8) |
| Three-phase PWM output (U phase/UB phase, V phase/VB phase, W phase/WB phase) | 13 channels (master channel: 1, slave channel: 12) |

Note: Above are examples of PWM output combinations.

Usages of each TAUDn channel are listed in the following table. CH2 is used as the master channel of CH3 to 9.

| TAUDn Channel | U phase / UB phase | V phase / VB phase | W phase / WB phase | Usage |
|---------------|--------------------|--------------------|--------------------|--|
| CH0 | — | — | — | Not used |
| CH1 | — | — | — | Not used |
| CH2 | √ | √ | √ | Carrier period (common to each phase) |
| CH3 | √ | √ | √ | Reserved |
| CH4 | √ | — | — | Delay pulse input (U phase/UB phase) |
| CH5 | √ | — | — | |
| CH6 | — | √ | — | Delay pulse input (V phase/VB phase) |
| CH7 | — | √ | — | |
| CH8 | — | — | √ | Delay pulse input (W phase/WB phase) |
| CH9 | — | — | √ | |
| CH10 | √ | — | — | U phase output (TOUT10) |
| CH11 | √ | — | — | UB phase output (TOUT11) |
| CH12 | — | √ | — | V phase output (TOUT12) |
| CH13 | — | √ | — | VB phase output (TOUT13) |
| CH14 | — | — | √ | W phase output (TOUT14) |
| CH15 | — | — | √ | WB phase output (TOUT15) |

Note: √: Used; —: Not used

(2) Configuration

The delay pulse output function with the dead time is realized by using the delay pulse output function/one-phase PWM output function of TAUDn and PIC1A in combination. The following figure shows the block diagram of the delay pulse output function with the dead time.

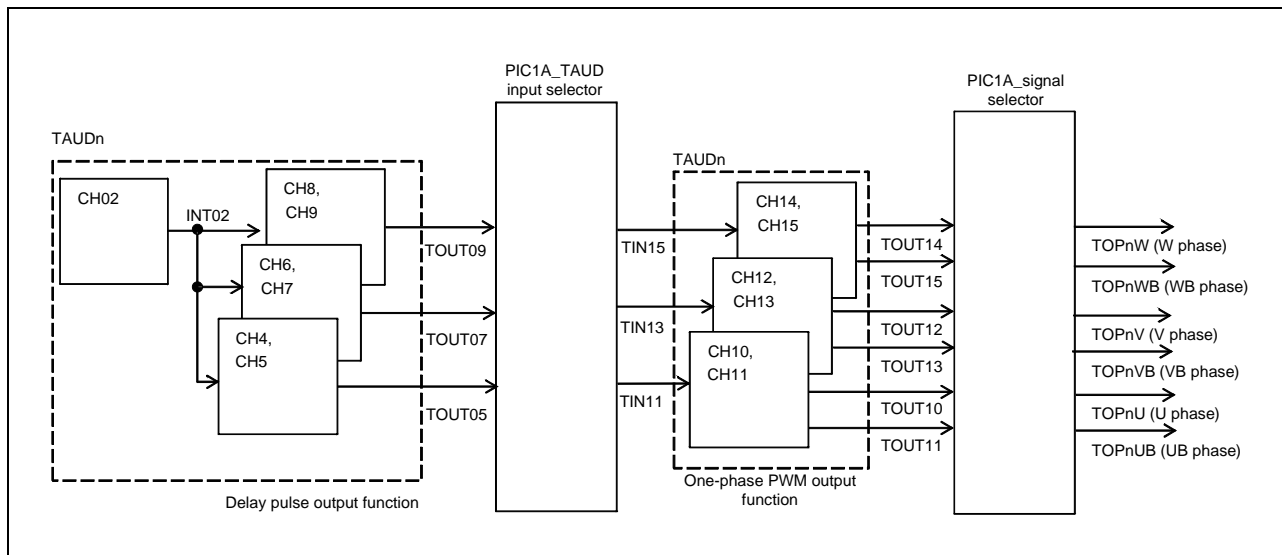


Figure 29.16 Block Diagram of Delay Pulse Output Function with Dead Time

The configuration of this function is described as follows using the PWM output of U phase/UB phase as an example.

- [TAUDn] Delay pulse output function
CH02, CH04, and CH05 are used in combination. The period, delay value, and the pulse width are set to CDR02, CDR04, and CDR05, respectively, and the delay pulse output signal (TOUT05) is generated.
- [PIC1A_TAUD input selector]
TOUT05 is selected for output as TIN11.
- [TAUDn] One-phase PWM output function
CH10 and CH11 are used in combination. The dead time value is set to CDR11, dead time is inserted in the PWM signal to be input to TIN11, and TOUT10 (U phase PWM signal) and TOUT11 (UB phase PWM signal) are output.
- [PIC1A_signal selector]
TOUT10 and TOUT11 inputs are selected and output to TOPnU and TOPnUB pins, respectively.

A similar process occurs for V phase/VB phase and W phase/WB phase.

(3) Registers

A block diagram of PIC1A is shown in the following figure.

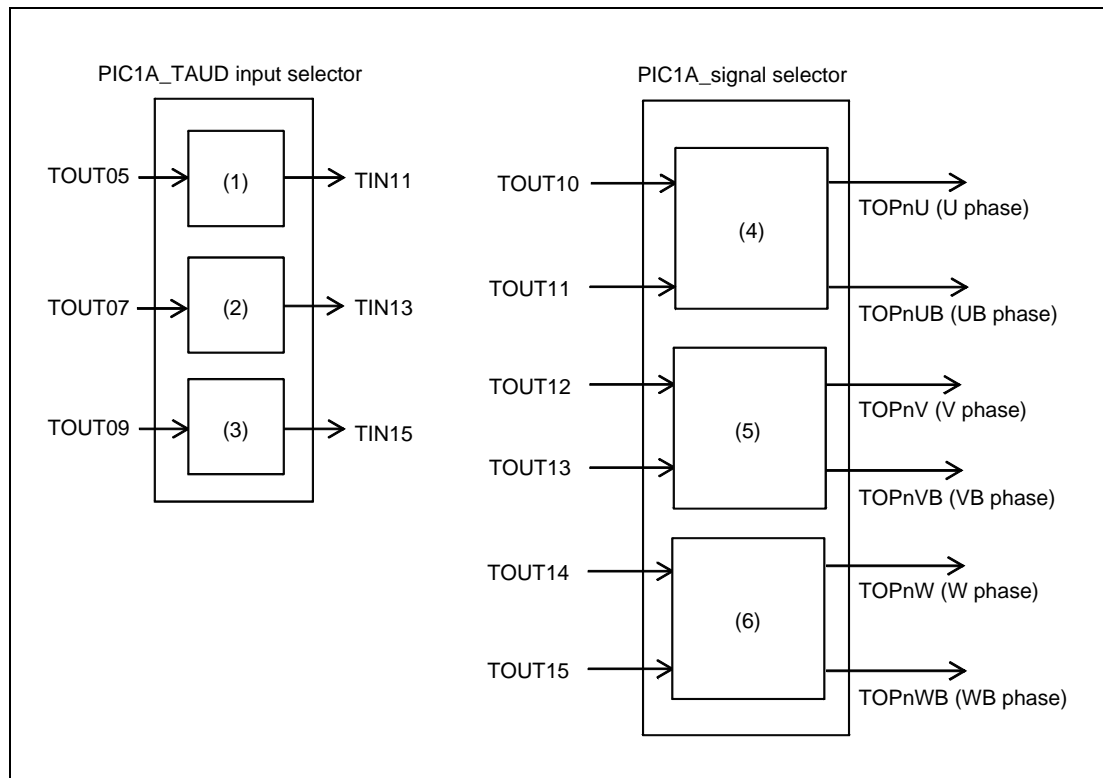


Figure 29.17 Block Diagram of PIC1A

The values of PIC1A registers used in this function are as follows.

U phase/ UB phase

The values to output TOUT05 as TIN11 (Figure 29.17, unit (1))

$$\text{PIC1AREG2n2}[19:18] = 10_{\text{B}}$$

$$\text{PIC1AREG2n2}[2] = 0_{\text{B}}$$

$$\text{PIC1ATAUDnSEL}[23:22] = 00_{\text{B}}$$

The values to output the TOUT10 and TOUT11 as TOPnU and TOPnUB, respectively (Figure 29.17, unit (4))

$$\text{PIC1AREG2n1}[19:16] = 0000_{\text{B}}$$

$$\text{PIC1AREG2n3}[2:0] = 000_{\text{B}}$$

$$\text{PIC1AREG2n3}[6:4] = 000_{\text{B}}$$

V phase/ VB phase

The values to output TOUT07 as TIN13 (**Figure 29.17**, unit (2))

$$\text{PIC1AREG2n2}[23:22] = 10_{\text{B}}$$

$$\text{PIC1AREG2n2}[3] = 0_{\text{B}}$$

$$\text{PIC1ATAUDnSEL}[27:26] = 00_{\text{B}}$$

The values to output TOUT12 and TOUT13 as TOPnV and TOPnVB, respectively (**Figure 29.17**, unit (5))

$$\text{PIC1AREG2n1}[23:20] = 0000_{\text{B}}$$

$$\text{PIC1AREG2n3}[10:8] = 000_{\text{B}}$$

$$\text{PIC1AREG2n3}[14:12] = 000_{\text{B}}$$

W phase/ WB phase

The values to output TOUT09 as TIN15 (**Figure 29.17**, unit (3))

$$\text{PIC1AREG2n2}[27:26] = 10_{\text{B}}$$

$$\text{PIC1AREG2n2}[4] = 0_{\text{B}}$$

$$\text{PIC1ATAUDnSEL}[31:30] = 00_{\text{B}}$$

The values to output TOUT14 and TOUT15 as TOPnW and TOPnWB, respectively (**Figure 29.17**, unit (6))

$$\text{PIC1AREG2n1}[27:24] = 0000_{\text{B}}$$

$$\text{PIC1AREG2n3}[18:16] = 000_{\text{B}}$$

$$\text{PIC1AREG2n3}[22:20] = 000_{\text{B}}$$

(4) Function

Details of this function are described using the delay pulse output with dead time (U phase/UB phase) as an example.

The following figure shows the timing diagram.

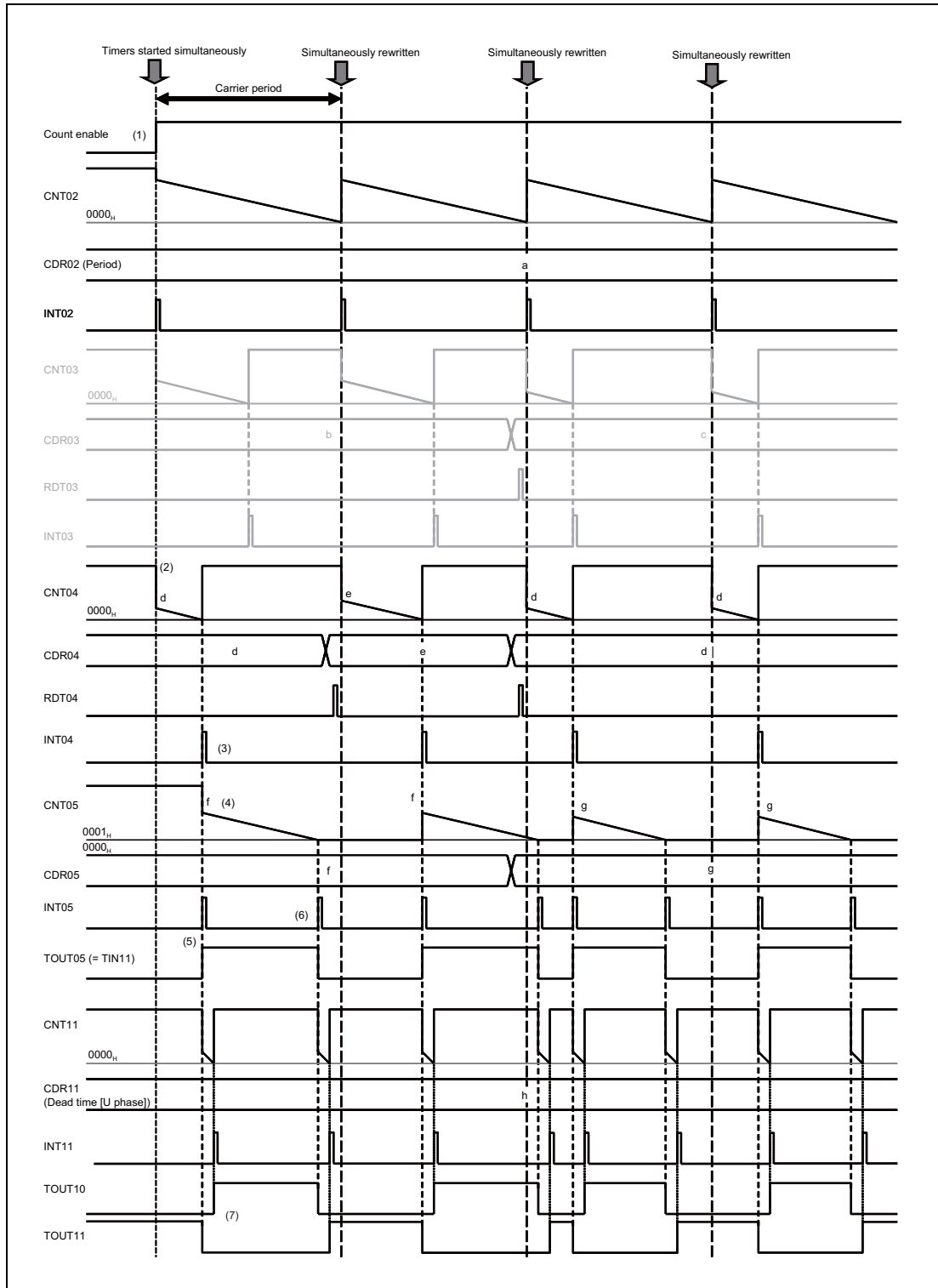


Figure 29.18 Timing Diagram of Delay Pulse Output with Dead Time (U phase/ UB phase)

- (1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.
- (2) For CH04, the set value is reloaded to CNT04 upon the occurrence of a CH02 underflow.
- (3) INT04 is generated upon the occurrence of a CH04 underflow.
- (4) The set value is reloaded to CDR05 upon the generation of INT04, causing CH05 to start operating.
- (5) When CH05 starts operating, INT05 is generated and the TOUT05 output level changes to the active level.
- (6) Upon CH05 underflow, INT05 is generated again and the TOUT05 output level changes to the inactive level. TOUT05 is supplied to TIN11.
- (7) The U phase PWM signal (TOUT10) and UB phase PWM signal (TOUT11) with the dead time are generated and output at the detection of the TIN11 edge, and output to TOPnU and TOPnUB.

A similar process occurs for V phase/VB phase and W phase/WB phase.

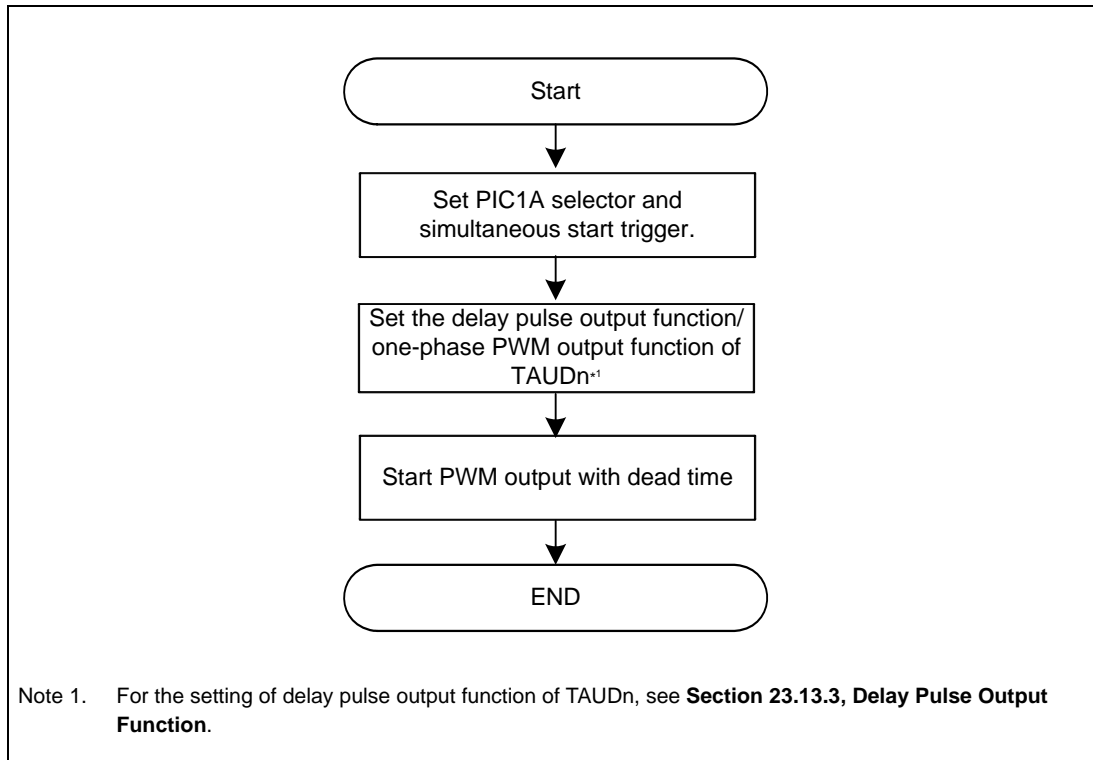
CAUTION

Do not set the delay value which extends over carrier period.

Select the count clock signal of the same clock for TAUDn.

(5) Flow Chart

The following flow chart shows the PWM output function with the dead time.



29.2.3.5 Trigger Pulse Interval Measurement Function

(1) Overview

This function allows measurement of trigger periods by inputting the trigger signal output from ENCA_n to TAUJ₀ and TAUD_n.

The following table lists the ENCA_n interrupt trigger signals to be measured by each timer and channel.

| Timer | Channel | Interrupt Signal to be Measured |
|-------|---------|---------------------------------|
| TAUJ0 | CH0 | ENCAT0IEC |
| | CH1 | ENCAT0IEC |
| | CH2 | ENCAT1IEC |
| | CH3 | ENCAT1IEC |
| TAUD0 | CH0 | ENCAT0EQ0 or ENCAT0EQ1 |
| | CH1 | ENCAT0EQ1 |
| | CH2 | ENCAT0EQ0 |
| TAUD1 | CH0 | ENCAT0EQ0 or ENCAT0EQ1 |
| | CH1 | ENCAT0EQ1 |
| | CH2 | ENCAT0EQ0 |

(2) Configuration

The trigger and pulse width measurement function is realized by using the TINm input pulse interval measurement functions of TAUJ0 and TAUDn, and PIC1A in combination. The following figure shows the block diagram of the trigger and pulse width measurement function.

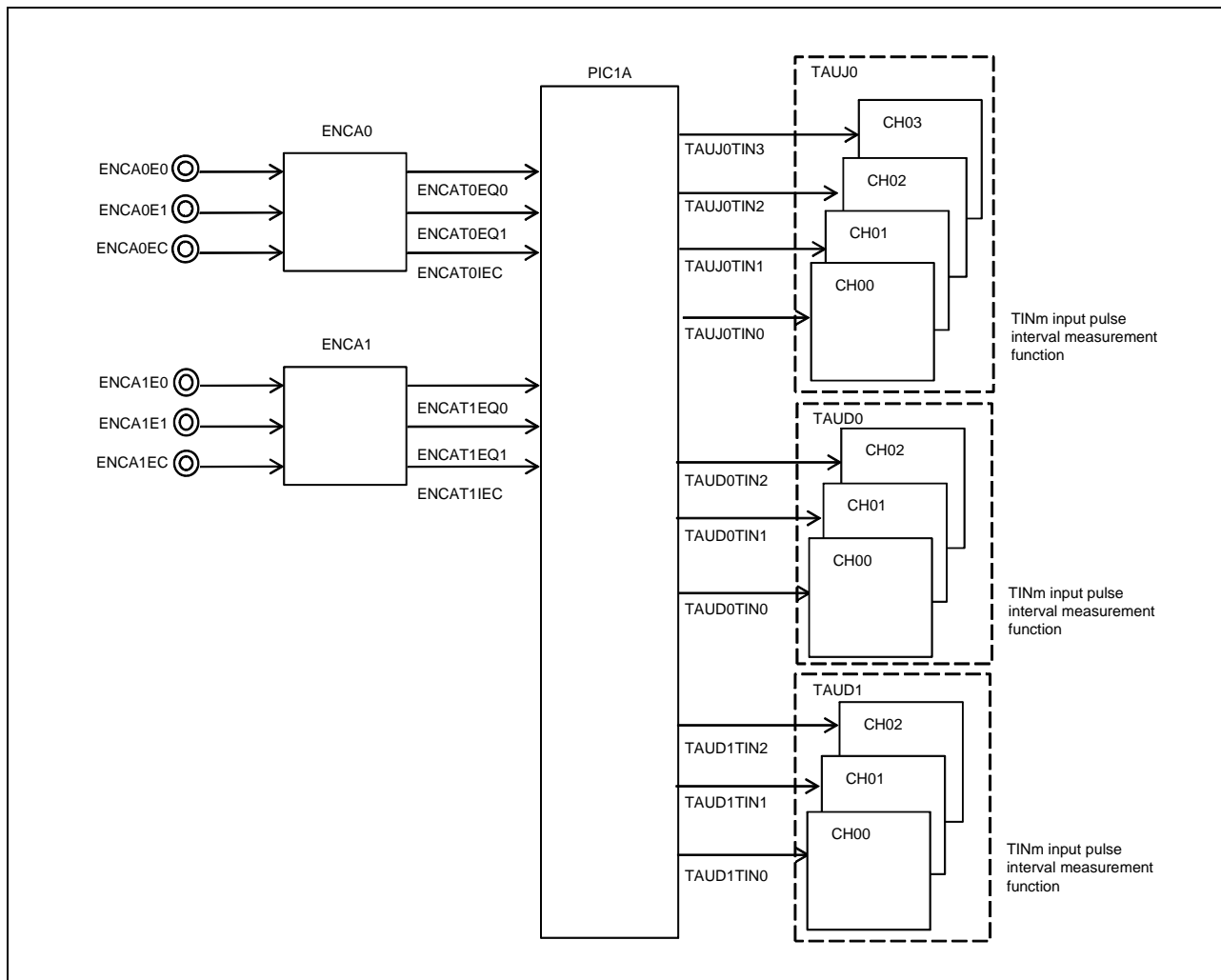


Figure 29.19 Block Diagram of Trigger and Pulse Width Measurement Function

The configuration of this function is described below using TAUJ0 CH0 as an example.

- [ENCA0]
ENCAT0IEC interrupt signal is generated each time ENCA0 timer counter is cleared by input from ENCA0EC pin.
- [PIC1A] Latch and toggle output (DT) circuit
ENCAT0IEC interrupt trigger signal selected by the DT circuit is converted into a level-sensitive toggle signal and output to TAUJ0TIN0.
- [TAUJ0] TINm input pulse interval measurement function
TAUJ0 CH0 is used. TAUJ0CNT0 is captured each time input signal is toggled. The counter is cleared and restarted.

Similar configuration is applied for trigger and pulse width measurement of TAUD0 and TAUD1.

(3) Registers

Block diagram of PIC1A is shown in the following figure.

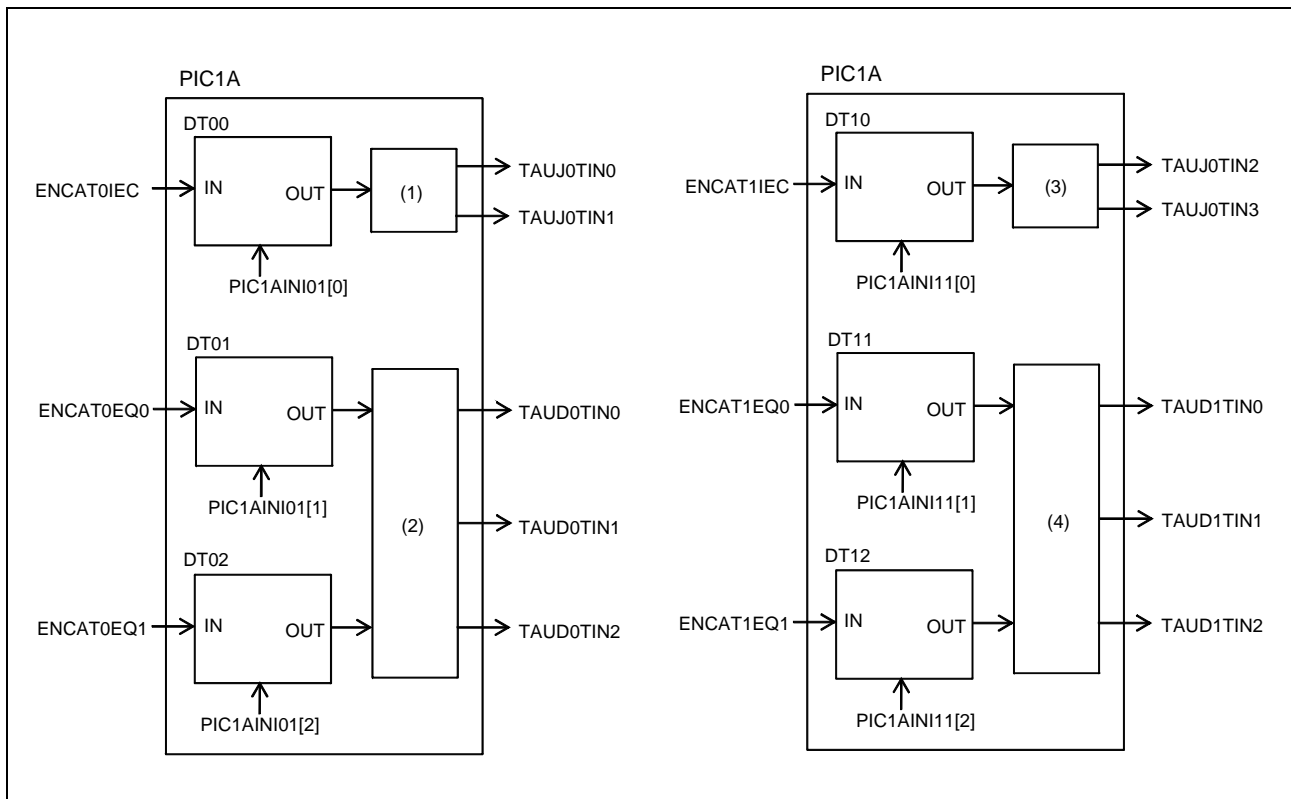


Figure 29.20 Block Diagram of PIC1A

The values of PIC1A registers used in this function are as follows.

ENCA0

- ENCAT0EC trigger and pulse width measurement
The register values to select the timer which performs ENCAT0EC trigger and pulse width measurement. (Figure 29.20, unit (1))

| Register Value | | TAUJ0.TIN00 | TAUJ0.TIN01 |
|----------------|---|---------------|-------------|
| PIC1AREG31 | | | |
| 1 | 0 | | |
| 0 | 0 | Not selected. | |
| 0 | 1 | ENCAT0IEC | — |
| 1 | 0 | — | ENCAT0IEC |
| 1 | 1 | ENCAT0IEC | ENCAT0IEC |

Note: Write 0 (value after reset) to PIC1AREG30[22,17:16]

- ENCAT0EQ0 and ENCAT0EQ1 trigger and pulse width measurement

The register values to select the timer which performs ENCAT0EQ0 and ENCAT0EQ1 trigger and pulse width measurement. (Figure 29.20, unit (2))

| Register Value | | | | | | | | TAUD0.TIN00 | TAUD0.TIN01 | TAUD0.TIN02 |
|----------------|----|----|----|---|---|---|---|---------------|-------------|-------------|
| PIC1AREG31 | | | | | | | | | | |
| 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Not selected. | | |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | ENCAT0EQ0 | ENCAT0EQ1 | — |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | ENCAT0EQ1 | — | ENCAT0EQ0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | ENCAT0EQ1 | ENCAT0EQ1 | ENCAT0EQ0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | ENCAT0EQ0 | ENCAT0EQ1 | ENCAT0EQ0 |

Note: Do not set the values other than those listed above for this function. Write 0 (value after reset) to PIC1ATAUD0SEL[5:0] and PIC1AREG30[22,17:16,1:0].

- Enable initialization of the DT02 to DT00 circuits

The register values to enable initialization of the DT02 to DT00 circuits.

PIC1AINI01[2:0] = 111_B (initialized)

ENCA1

ENCAT1EC trigger and pulse width measurement

The register values to select the timer which performs ENCAT1EC trigger and pulse width measurement. (Figure 29.20, unit (3))

| Register Value | | | | TAUJ0.TIN02 | TAUJ0.TIN03 |
|----------------|--|---|--|---------------|-------------|
| PIC1AREG31 | | | | | |
| 4 | | 3 | | | |
| 0 | | 0 | | Not selected. | |
| 0 | | 1 | | ENCAT1IEC | — |
| 1 | | 0 | | — | ENCAT1IEC |
| 1 | | 1 | | ENCAT1IEC | ENCAT1IEC |

Note: Write 0 (value after reset) to PIC1AREG30[20:19,11:10]

- ENCAT1EQ0 and ENCAT1EQ1 trigger and pulse width measurement

The register values to select the timer which performs ENCAT1EQ0 and ENCAT1EQ1 trigger and pulse width measurement. (Figure 29.20, unit (4))

| Register Value | | | | | | | | TAUD1.TIN00 | TAUD1.TIN01 | TAUD1.TIN02 |
|----------------|----|----|----|----|----|----|----|---------------|-------------|-------------|
| PIC1AREG31 | | | | | | | | | | |
| 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Not selected. | | |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | ENCAT1EQ0 | ENCAT1EQ1 | — |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | ENCAT1EQ1 | — | ENCAT1EQ0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | ENCAT1EQ1 | ENCAT1EQ1 | ENCAT1EQ0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | ENCAT1EQ0 | ENCAT1EQ1 | ENCAT1EQ0 |

Note: Do not set the values other than those listed above for this function. Write 0 (value after reset) to PIC1ATAUD1SEL[5:0] and PIC1AREG30[20:19,9:6].

- Enable initialization of DT12 to DT10 circuits

The register values to enable initialization of the DT12 to DT10 circuits.

PIC1AINI11[2:0] = 111_B (initialized)

(4) Function

Details of this function are described here.

The following figure shows the timing diagram.

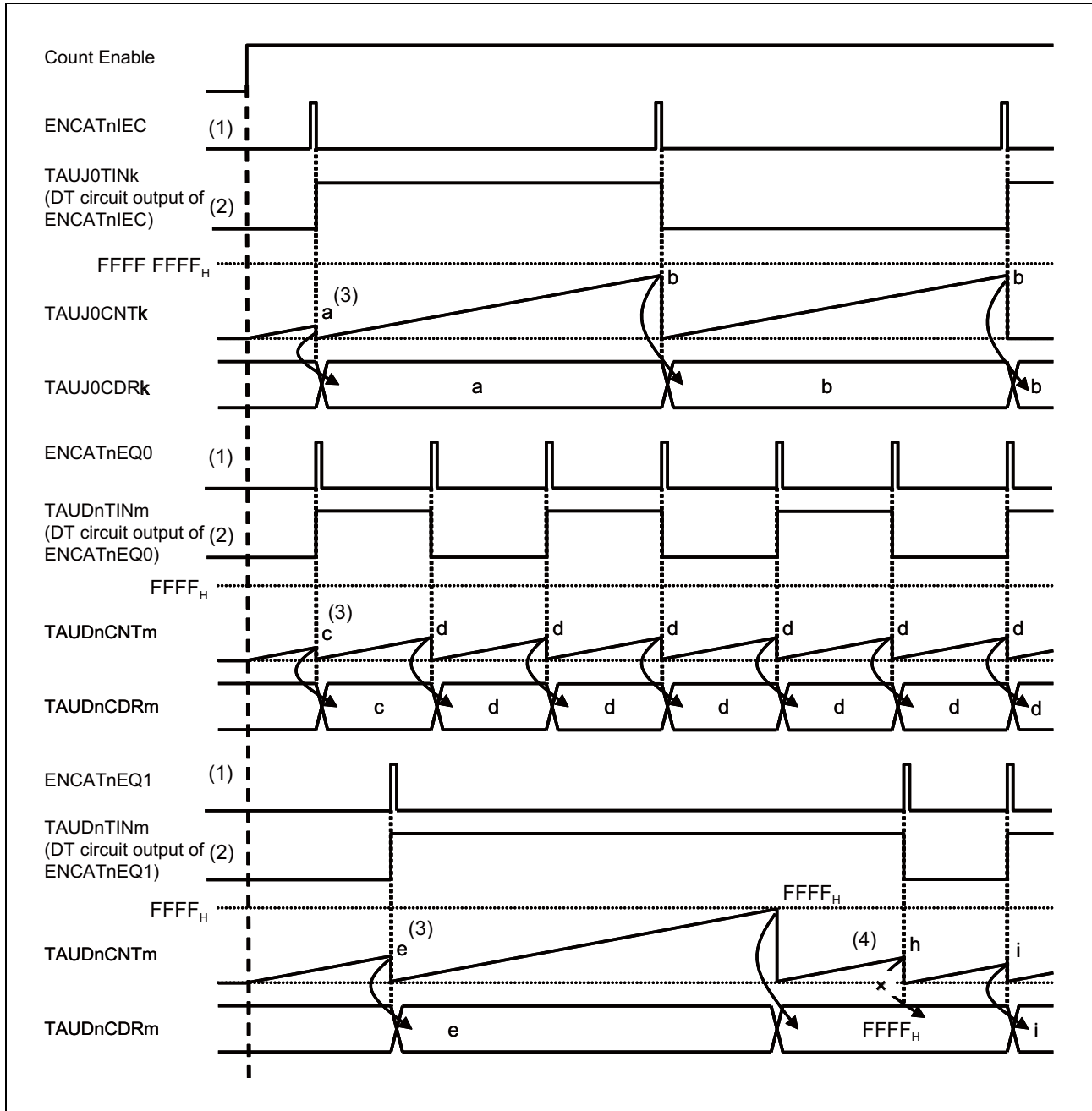


Figure 29.21 Timing Diagram of Trigger and Pulse Width Measurement

- (1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.
- (2) The interrupt trigger signal output from ENCA_n is converted to the level-sensitive toggle signal by the DT circuit and is output to TIN_m of TAUJ₀ and TIN_m of TAUD_n.
- (3) The CNT_m value is captured into CDR_m on the TIN_m toggle timing and the counter is cleared at the same time.
- (4) When an overflow occurs, the greatest count value (FFFF_H for TAUD_n and FFFF FFFF_H for TAUJ₀) is captured and the counter is cleared at the same time. The CNT_m value is not captured on the first trigger after the overflow. (When TAUD_nCMOR_m.TAUD_nCOS[1] = 1_B)

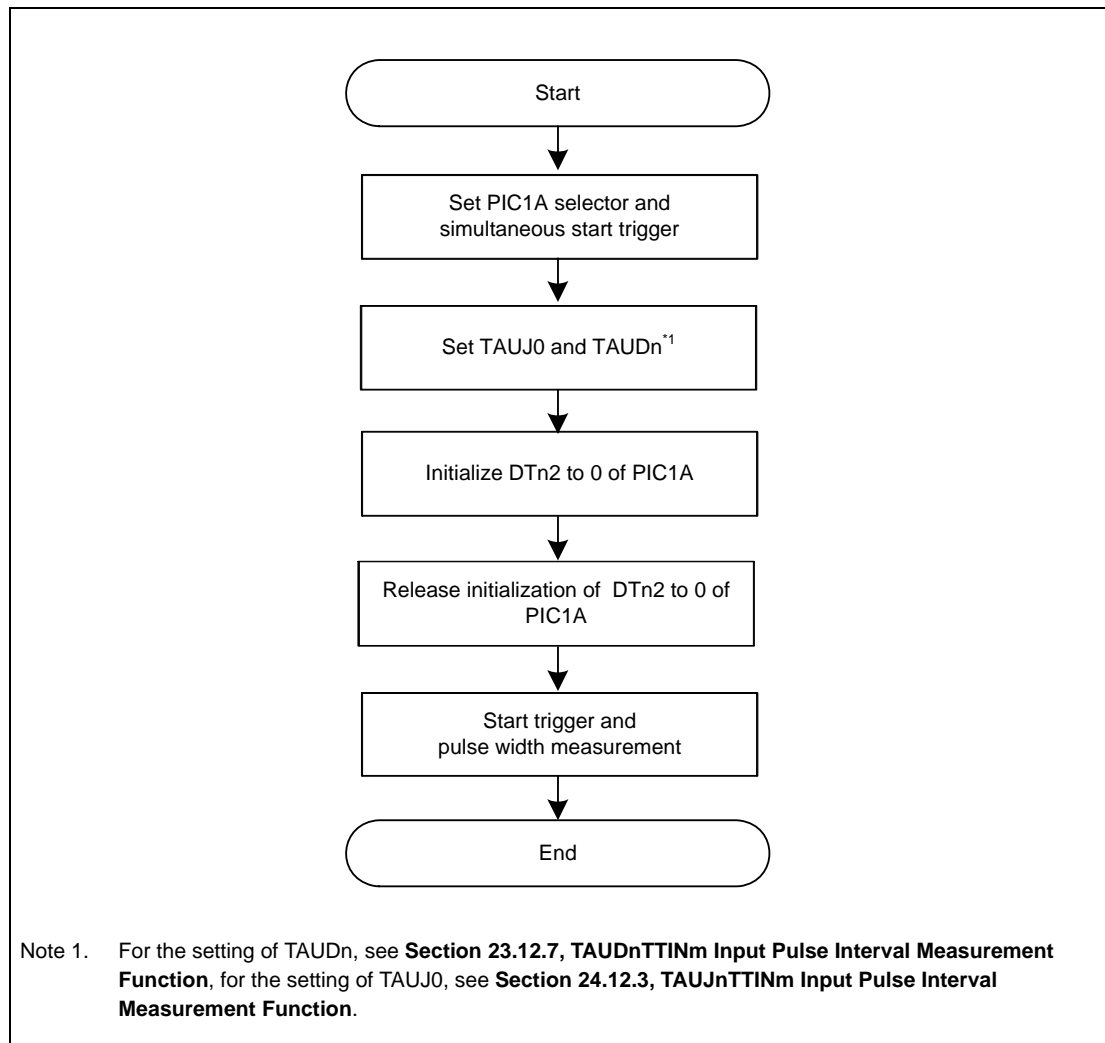
CAUTION

Operation at an overflow varies depending on the setting of TAUJ and TAUD. For details of the TAUJ setting, see Section 24.12.3, TAUJ_nTTIN_m Input Pulse Interval Measurement Function, see Section 23.12.7, TAUD_nTTIN_m Input Pulse Interval Measurement Function. In this function, set both edges (rising and falling) as the valid edge of TIN_m of TAUJ₀ and TAUD_n to be detected.

(5) Flow Chart

The following figure shows the setting flow of this function.

These settings can be performed while ENCA_n is operating as well as while waiting for the simultaneous start trigger.



The ENCA_n settings to use this function are as follows.

ENCA_nCTL[15:0] = xx00 0000 x00x xxxx_B

ENCA_nIOC0[7:0] = 0000 0000_B

ENCA_nIOC1[7:0] = (set any value)

“x” can be set arbitrarily. For the registers specifications, see **Section 28, Encoder Timer (ENCA)**.

29.2.3.6 Encoder Capture Trigger Select Function

(1) Overview

This function selects either ADcGnTRGm (ADcGn conversion start trigger signal m), TAUDnTINTm (TAUDn-CHm interrupt signal), or ENcAnI1 (ENcAn external pin input signal) as the ENcAn capture trigger signal.

(2) Configuration

The encoder capture trigger select function is realized by using ADcGnTRGm (ADcGn conversion start trigger signal m), TAUDnTINTm (TAUDn CHm interrupt signal), ENcAnI1 (ENcAn external pin input 1 signal), and PIC1A in combination.

Block diagram of this function is shown in the following figure.

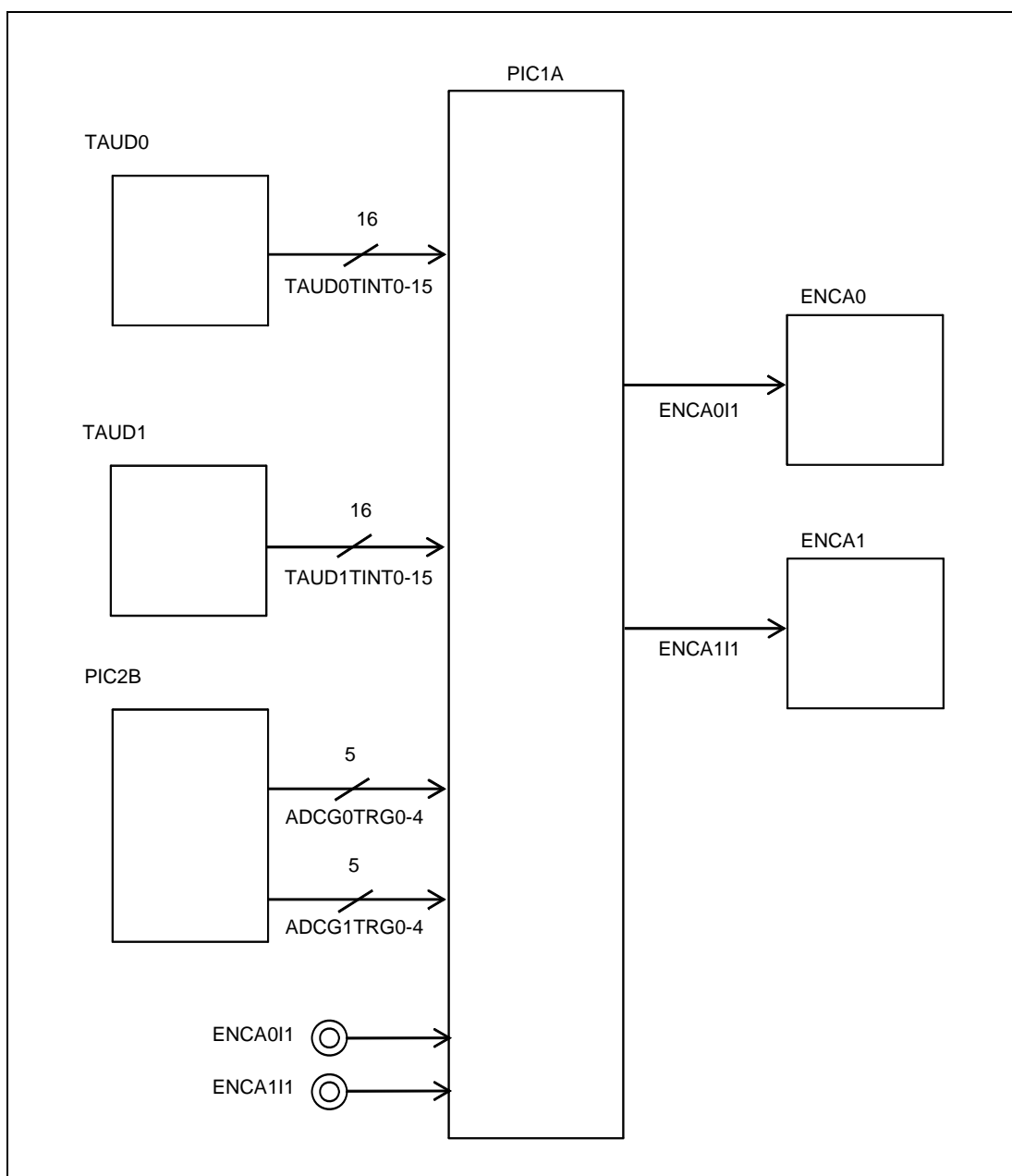


Figure 29.22 Block Diagram of Encoder Capture Trigger Select Function

An example of selecting CH0 of TAUD0 as a capture trigger input of ENCA0 is described below.

$$\text{PIC1AENCSEL400}[7] = 1_{\text{B}}$$

$$\text{PIC1AENCSEL400}[3:0] = 0000_{\text{B}}$$

$$\text{PIC1AREG30}[18] = 1_{\text{B}}$$

$$\text{PIC1AREG30}[5:2] = 0000_{\text{B}}$$

(3) Registers

Block diagram of PIC1A is shown in the following figure.

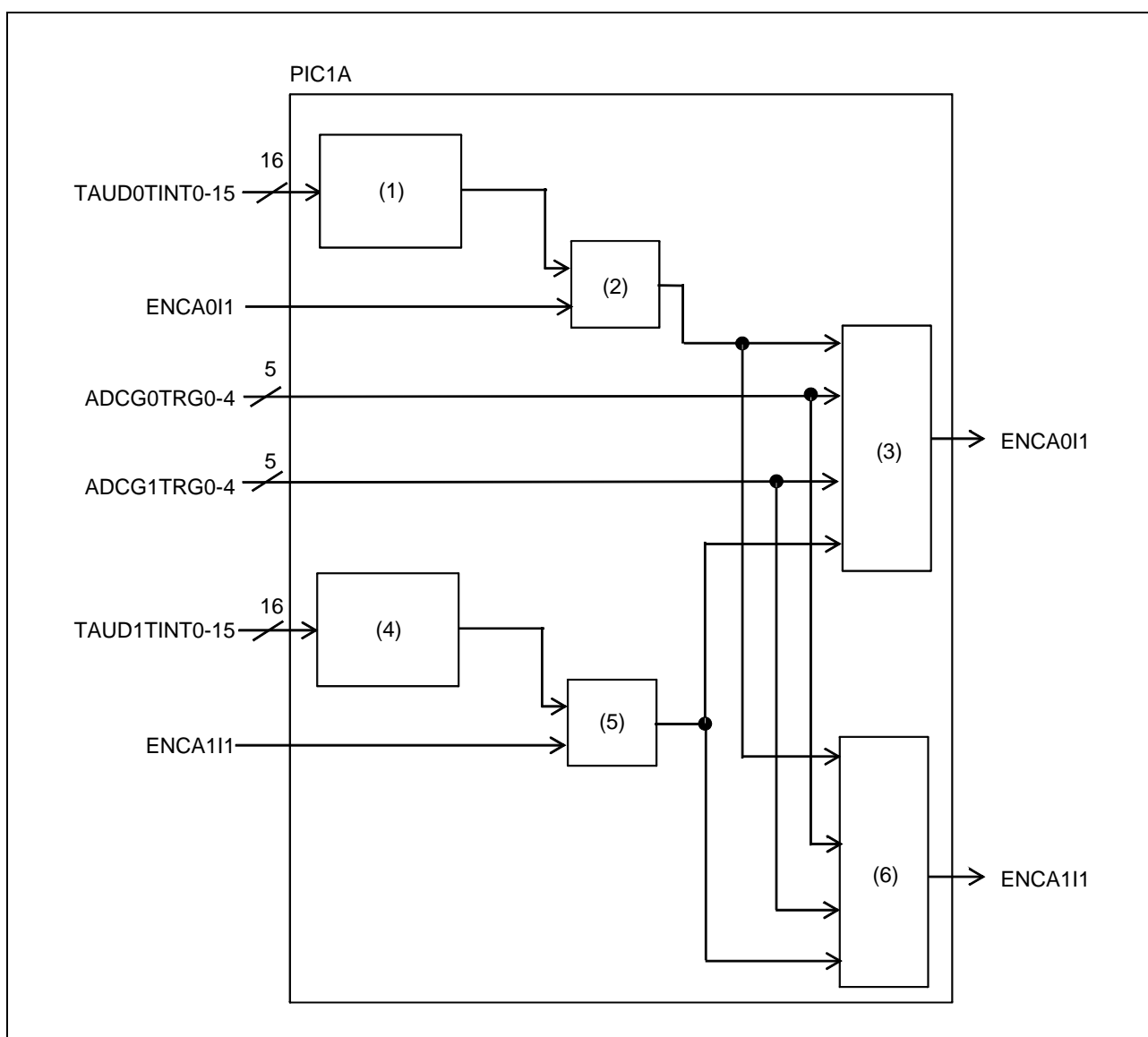


Figure 29.23 Block Diagram of PIC1A

The values of PIC1A registers used in this function are as follows.

ENCA0

(1) TAUD0TINTm selection

The register values to select TAUD0TINTm. Set 1 to the PIC1AENCSEL400[7] to select TAUD0TINTm.

| Register Values | | | | | Output of (1) |
|-----------------|---|---|---|---|---------------|
| PIC1AENCSEL400 | | | | | |
| 7 | 3 | 2 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | INTTAUD0I0 |
| 1 | 0 | 0 | 0 | 1 | INTTAUD0I1 |
| 1 | 0 | 0 | 1 | 0 | INTTAUD0I2 |
| 1 | 0 | 0 | 1 | 1 | INTTAUD0I3 |
| 1 | 0 | 1 | 0 | 0 | INTTAUD0I4 |
| 1 | 0 | 1 | 0 | 1 | INTTAUD0I5 |
| 1 | 0 | 1 | 1 | 0 | INTTAUD0I6 |
| 1 | 0 | 1 | 1 | 1 | INTTAUD0I7 |
| 1 | 1 | 0 | 0 | 0 | INTTAUD0I8 |
| 1 | 1 | 0 | 0 | 1 | INTTAUD0I9 |
| 1 | 1 | 0 | 1 | 0 | INTTAUD0I10 |
| 1 | 1 | 0 | 1 | 1 | INTTAUD0I11 |
| 1 | 1 | 1 | 0 | 0 | INTTAUD0I12 |
| 1 | 1 | 1 | 0 | 1 | INTTAUD0I13 |
| 1 | 1 | 1 | 1 | 0 | INTTAUD0I14 |
| 1 | 1 | 1 | 1 | 1 | INTTAUD0I15 |

(2) TAUD0TINTm and ENCA0I1 pins selection

The register values to select either the output (1) or ENCA0I1.

| Register Values | | Output (2) |
|-----------------|--|------------|
| PIC1AREG30 | | |
| 18 | | |
| 1 | | Output (1) |
| 0 | | ENCA0I1 |

(3) ENCA0I1 selection

The register values to select any of the output (2), output (5), ADCG0TRG0-4, or ADCG1TRG0-4.

| Register Values | | | | ENCA0I1 |
|-----------------|---|---|---|------------|
| PIC1AREG30 | | | | |
| 5 | 4 | 3 | 2 | |
| 0 | 0 | 0 | 0 | Output (2) |
| 0 | 0 | 0 | 1 | Output (5) |
| 0 | 0 | 1 | 0 | ADCG0TRG4 |
| 0 | 0 | 1 | 1 | ADCG0TRG3 |
| 0 | 1 | 0 | 0 | ADCG0TRG2 |
| 0 | 1 | 0 | 1 | ADCG0TRG1 |
| 0 | 1 | 1 | 0 | ADCG0TRG0 |
| 0 | 1 | 1 | 1 | ADCG1TRG4 |
| 1 | 0 | 0 | 0 | ADCG1TRG3 |
| 1 | 0 | 0 | 1 | ADCG1TRG2 |
| 1 | 0 | 1 | 0 | ADCG1TRG1 |
| 1 | 0 | 1 | 1 | ADCG1TRG0 |

Note: Do not set the values other than those listed above for this function.

ENCA1

(4) TAUD1TINTm selection

The register values to select TAUD1TINTm. Set 1 to the PIC1AENCSEL410[7] to select TAUD1TINTm.

| Register Values | | | | | Output (4) |
|-----------------|---|---|---|---|-------------|
| PIC1AENCSEL410 | | | | | |
| 7 | 3 | 2 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | INTTAUD110 |
| 1 | 0 | 0 | 0 | 1 | INTTAUD111 |
| 1 | 0 | 0 | 1 | 0 | INTTAUD112 |
| 1 | 0 | 0 | 1 | 1 | INTTAUD113 |
| 1 | 0 | 1 | 0 | 0 | INTTAUD114 |
| 1 | 0 | 1 | 0 | 1 | INTTAUD115 |
| 1 | 0 | 1 | 1 | 0 | INTTAUD116 |
| 1 | 0 | 1 | 1 | 1 | INTTAUD117 |
| 1 | 1 | 0 | 0 | 0 | INTTAUD118 |
| 1 | 1 | 0 | 0 | 1 | INTTAUD119 |
| 1 | 1 | 0 | 1 | 0 | INTTAUD1110 |
| 1 | 1 | 0 | 1 | 1 | INTTAUD1111 |
| 1 | 1 | 1 | 0 | 0 | INTTAUD1112 |
| 1 | 1 | 1 | 0 | 1 | INTTAUD1113 |
| 1 | 1 | 1 | 1 | 0 | INTTAUD1114 |
| 1 | 1 | 1 | 1 | 1 | INTTAUD1115 |

(5) TAUD1TINTm and ENCA1I1 pins selection

The register values to select either the output (4) or ENCA1I1.

| Register Values | | Output (5) |
|-----------------|--|------------|
| PIC1AREG30 | | |
| 21 | | |
| 1 | | Output (4) |
| 0 | | ENCA1I1 |

(6) ENCA1I1 selection

The register values to select any of the output (2), output (5), ADCG0TRG0-4, or ADCG1TRG0-4.

| Register Values | | | | ENCA1I1 |
|-----------------|----|----|----|------------|
| PIC1AREG30 | | | | |
| 15 | 14 | 13 | 12 | |
| 0 | 0 | 0 | 0 | Output (5) |
| 0 | 0 | 0 | 1 | Output (2) |
| 0 | 0 | 1 | 0 | ADCG0TRG4 |
| 0 | 0 | 1 | 1 | ADCG0TRG3 |
| 0 | 1 | 0 | 0 | ADCG0TRG2 |
| 0 | 1 | 0 | 1 | ADCG0TRG1 |
| 0 | 1 | 1 | 0 | ADCG0TRG0 |
| 0 | 1 | 1 | 1 | ADCG1TRG4 |
| 1 | 0 | 0 | 0 | ADCG1TRG3 |
| 1 | 0 | 0 | 1 | ADCG1TRG2 |
| 1 | 0 | 1 | 0 | ADCG1TRG1 |
| 1 | 0 | 1 | 1 | ADCG1TRG0 |

Note: Do not set the values other than those listed above for this function.

(4) Function

Function

Details of this function are described with an example of selecting TAUDnTINTm as a capture trigger signal.

The following figure shows the timing diagram.

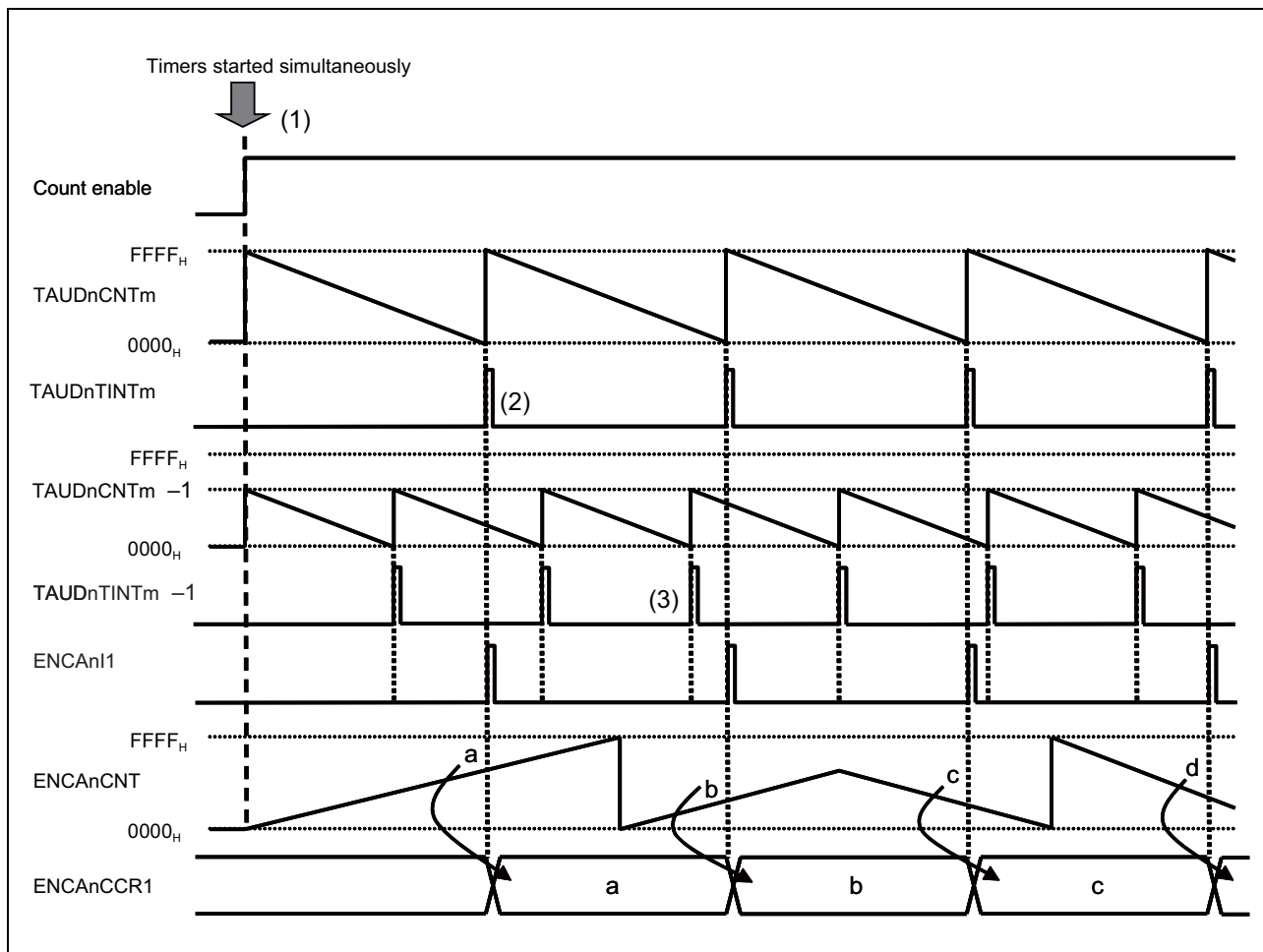


Figure 29.24 Timing Diagram of Encoder Capture Trigger Select Function (TAUDnTINTm)

- (1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.
- (2) On generation of an effective edge of TAUDnTINTm, ENCAAn captures ENCAAnCNT.

Do not select ENCAAn interrupt trigger signal (INTENCATnI1) as the ADCGn trigger described in **Section 29.3.3.1, ADCGn Trigger Select Function**. If it is selected, the correct operation cannot be performed because the following loop occurs: ADCGnTRG1 generation → ENCAAn capture operation → INTENCATnI1 generation by capture operation → ADCGnTRG1 generation.

The following figure shows the loop paths of PIC1A, PIC2B, and ENCA_n.

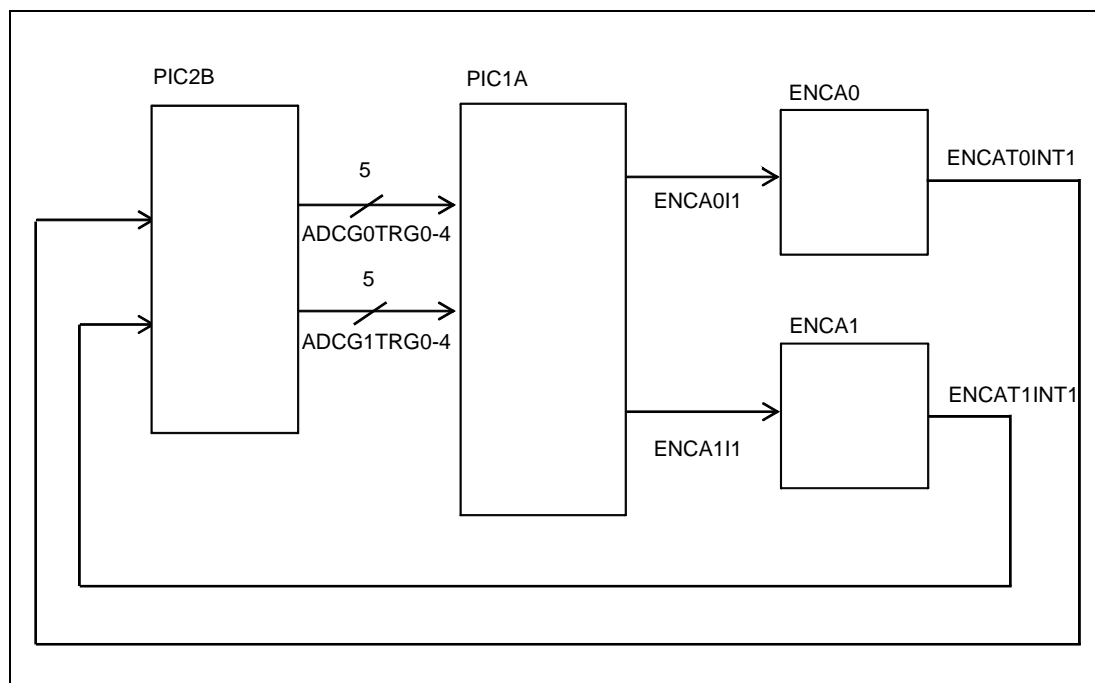


Figure 29.25 Timing Chart of Loop Paths of PIC1A, PIC2B, and ENCA_n

(5) Flow Chart

Select the encoder capture trigger before starting the encoder timer.

ENCA_n settings to use this function are as follows.

$$\text{ENCA}_n\text{CTL}[15:0] = 0000\ 001x\ 0\ 00x\ \text{xxxx}_B$$

$$\text{ENCA}_n\text{IOC0}[7:0] = 0000\ 01x_x_B$$

$$\text{ENCA}_n\text{IOC1}[7:0] = (\text{set any value})$$

“x” can be set arbitrarily. For the registers specifications, see **Section 28, Encoder Timer (ENCA)**.

29.2.3.7 Two-Phase Encoder Control Function (Control Method 1)

(1) Overview

This function allows switching of the output patterns of the motor control function (TSG3n) in 120-DC mode using the two-phase encoder control function (ENCA_n).

(2) Configuration

Switching of output pattern in 120-DC mode by encoder result is realized by using ENCA_n and TSG3n, and PIC1A in combination. The following figure describes the block diagram of two-phase encoder control function (control method 1).

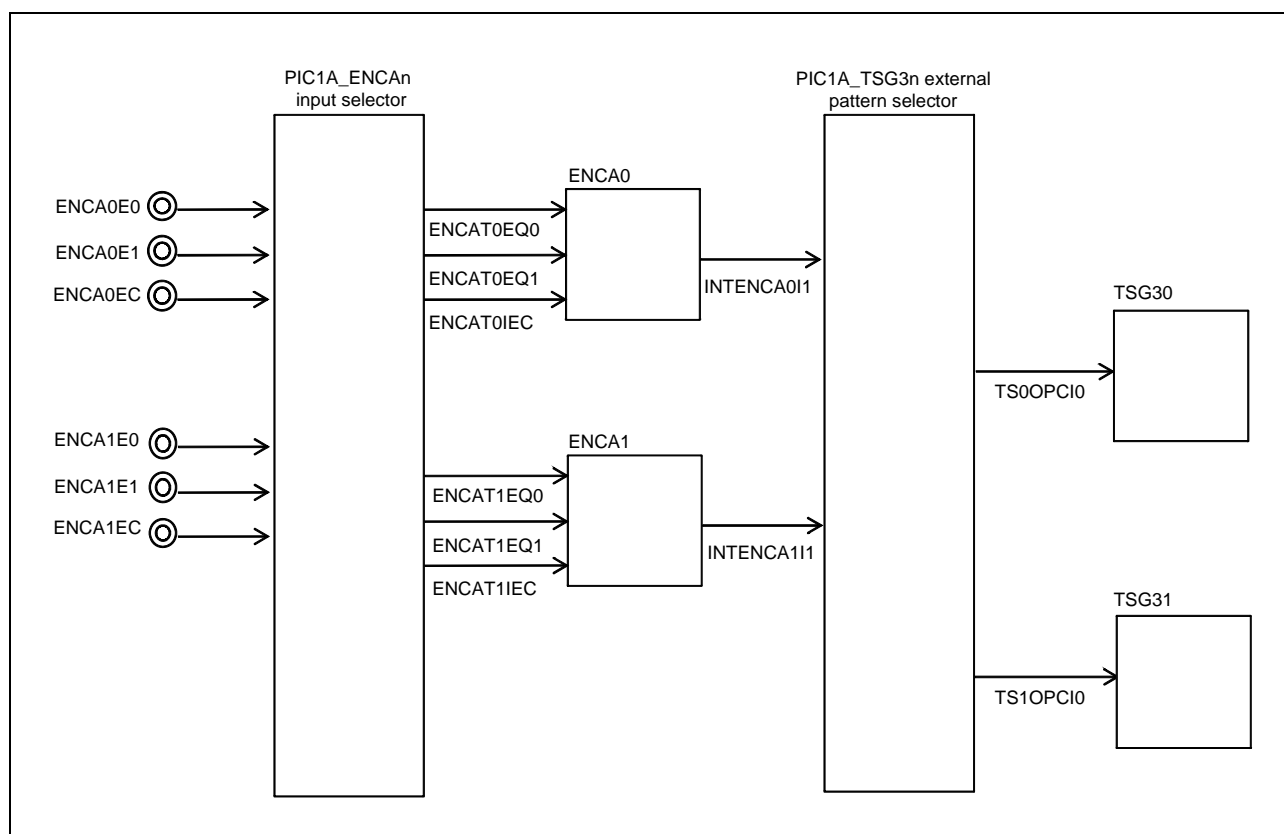


Figure 29.26 Block Diagram of Two-Phase Encoder Control Function (Control Method 1)

The configuration of two-phase encoder control function (control method 1) is described below.

- [PIC1A_ENCAn input selector]
ENCA_nE0, ENCA_nE1, and ENCA_nEC pin inputs are selected and output as ENCA_nEQ0, ENCA_nEQ1, and ENCA_nIEC.
- [ENCA_n]
INTENCA_nI1 is output by two-phase encoder processing.
- [PIC1A_TSG3n external pattern selector]
INTENCA_nI1 is selected and output to TS0OPCI0 or TS1OPCI0.
- [TSG3n]
Output pattern in 120-DC mode is switched by TSG3nOPCI0.

(3) Registers

The values of PIC1A registers used in this function are as follows.

PIC1A_ENCA_n input selector

The register values to output ENCA_n pin inputs (ENCA_nE0, ENCA_nE1, and ENCA_nEC) as ENCA_nEQ0, ENCA_nEQ1, and ENCA_nIEC.

$$\text{PIC1AREG30}[22] = 0_{\text{B}}$$

$$\text{PIC1AREG30}[20:19] = 00_{\text{B}}$$

$$\text{PIC1AREG30}[17:16] = 00_{\text{B}}$$

$$\text{PIC1AREG30}[11:6] = 000000_{\text{B}}$$

$$\text{PIC1AREG30}[1:0] = 00_{\text{B}}$$

PIC1A_TSG3_n external pattern selection

The register values to select the interrupt signal to be input as TSG30 external pattern.

| Register Values | | | | TS0OPCI0 |
|-----------------|---|---|---|------------|
| PIC1AREG50 | | | | |
| 10 | 8 | 6 | 5 | |
| X | 0 | 0 | 1 | INTENCA011 |
| 0 | X | 1 | 0 | INTENCA111 |

Note: Do not set the values other than those listed above for this function.

The register values to select the interrupt signal to be input as TSG31 external pattern.

| Register Values | | | | TS1OPCI0 |
|-----------------|---|---|---|------------|
| PIC1AREG51 | | | | |
| 10 | 8 | 6 | 5 | |
| X | 0 | 0 | 1 | INTENCA011 |
| 0 | X | 1 | 0 | INTENCA111 |

Note: Do not set the values other than those listed above for this function.

(4) Function

Details of this function are described using the two-phase encoder control function (method 1) at up count (normal rotation) as an example.

The following figure shows the timing diagram.

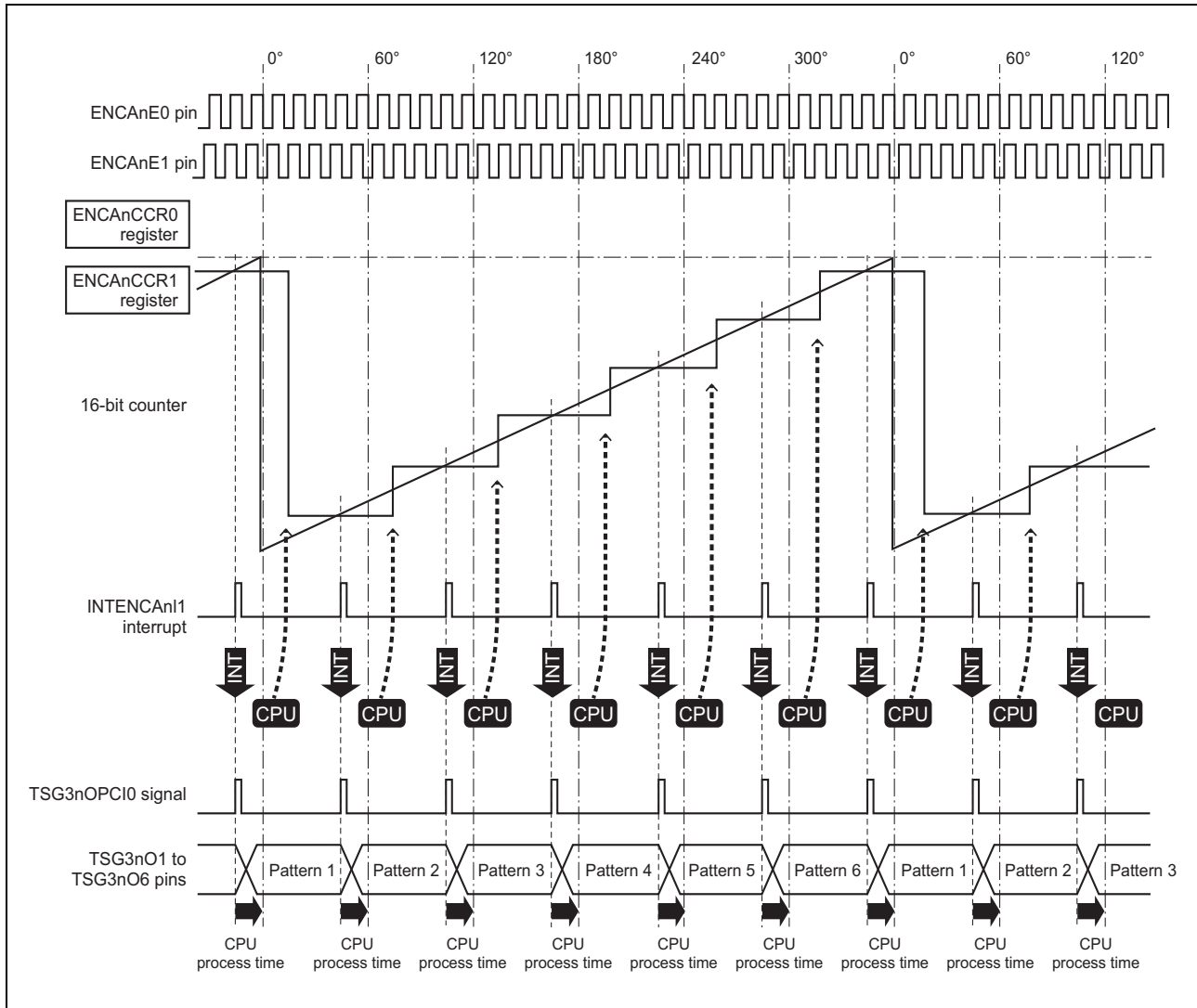


Figure 29.27 Timing Diagram of Two-Phase Encoder Control Function (Control Method 1) at Up Count (Normal Rotation)

- (1) The encoder counter value matches the ENCA_nCCR1, INTENCA_nI1 is generated, and the set pattern is output from the TSG3nO1 to TSG3nO6 pins.
- (2) CPU calculates the next timing to switch output patterns by an interrupt processing and sets the value to ENCA_nCCR1.
- (3) Upon a match of the encoder counter value and ENCA_nCCR0, the encoder counter is cleared.

CAUTION

It is necessary to set ENCA_nCCR1 at each pattern switch (each INTENCA_n1 interrupt). It is necessary to match the initial output pattern of timer TSG3_n to the set value of ENCA_nCCR1 before starting, because this value is not cleared by the encoder clear input.

Switching between normal and reverse rotations of output patterns is set with the TSG3_nPSC bit of the TSG3_nOPT0 register.

The following figure shows the timing diagram at down count (reverse rotation).

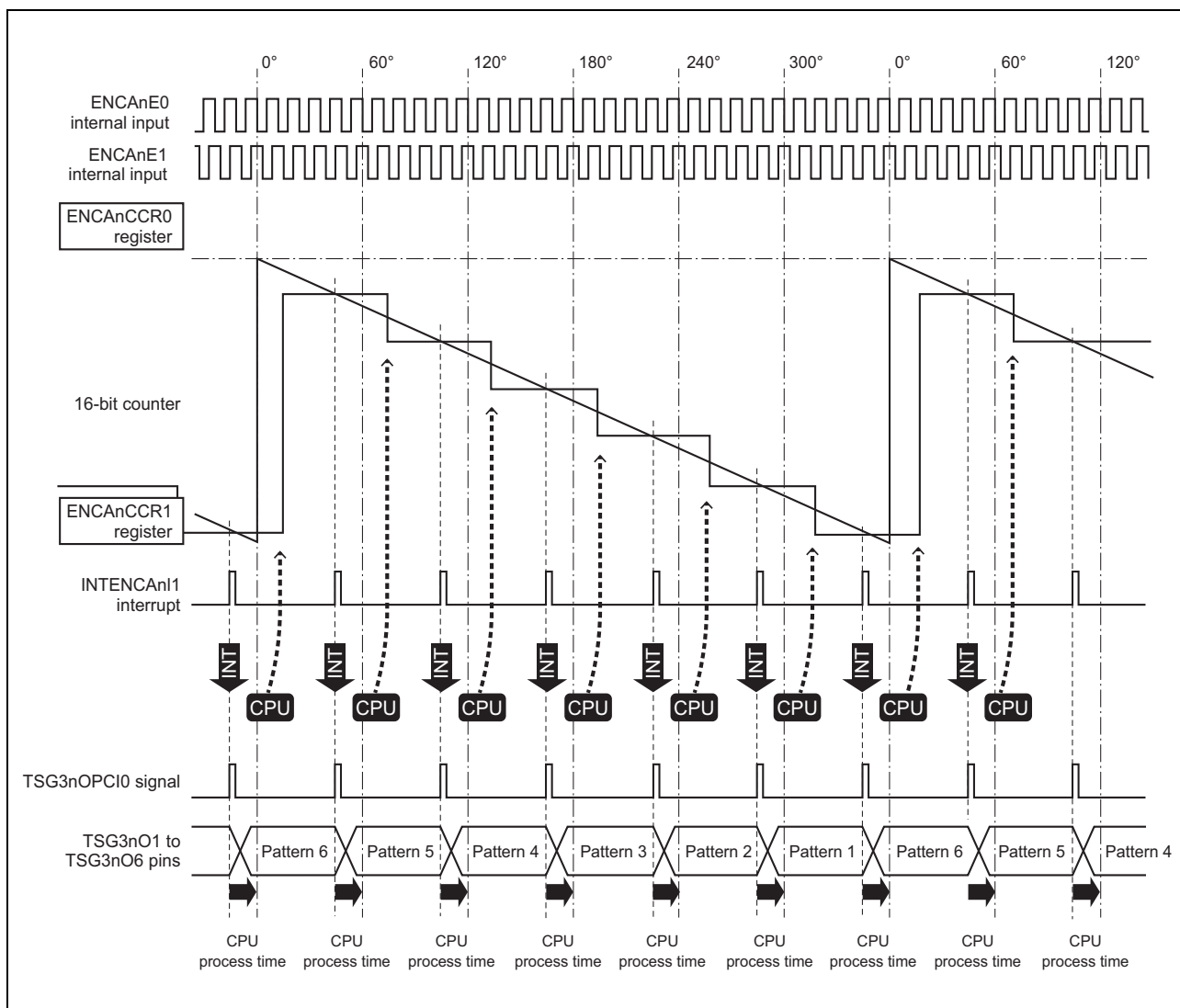
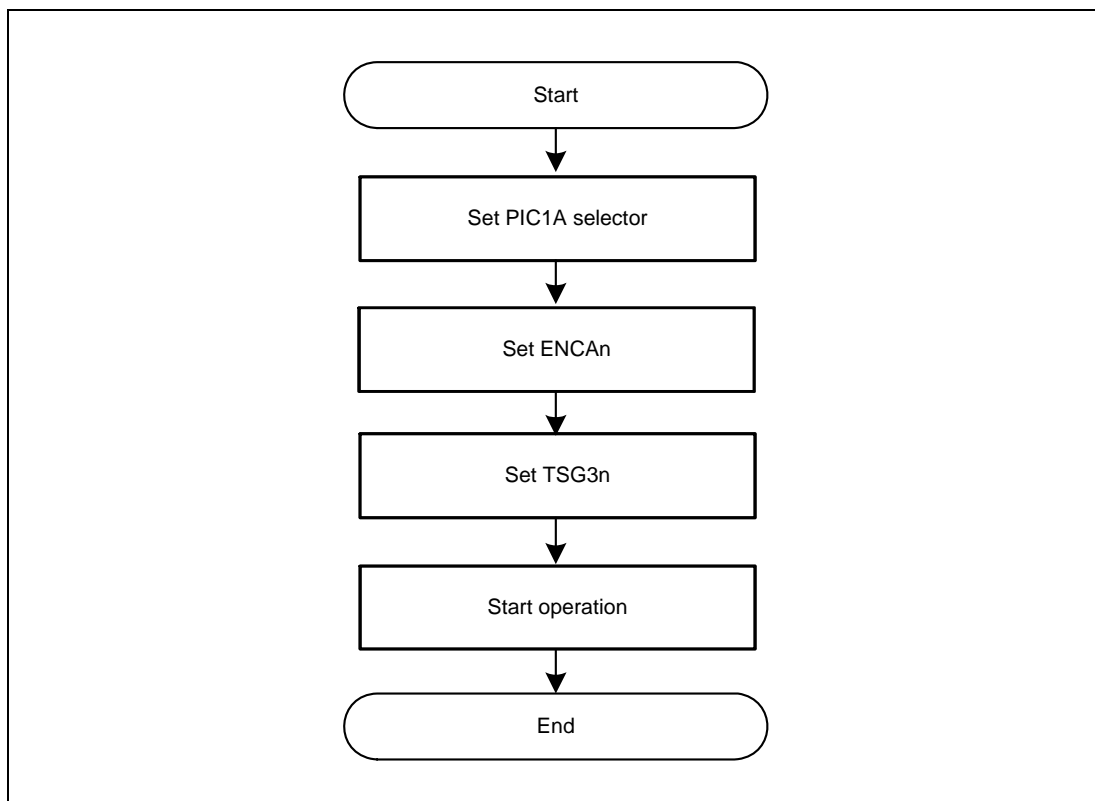


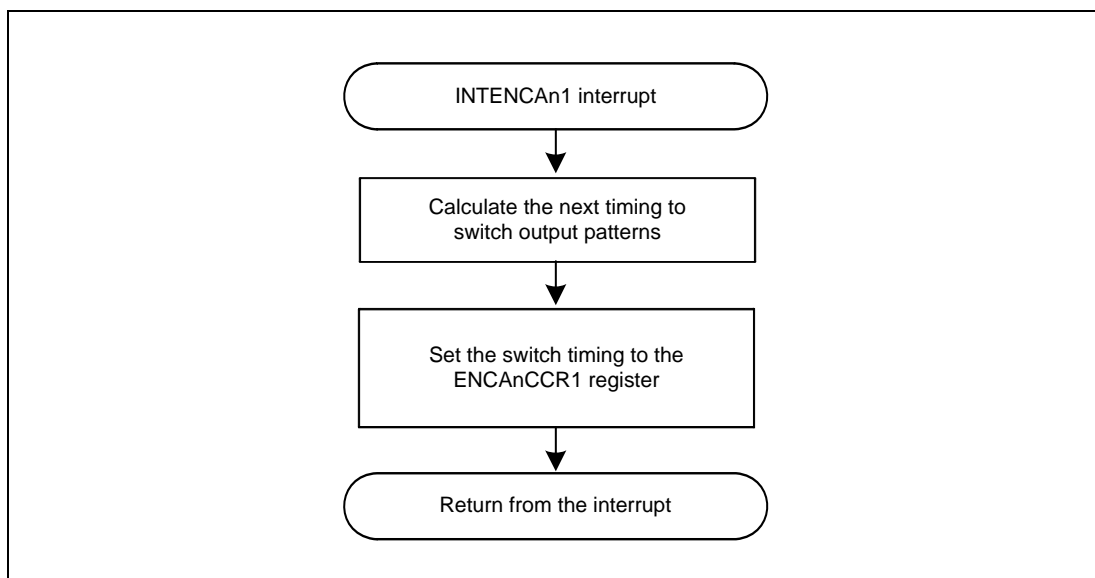
Figure 29.28 Timing Diagram of Two-Phase Encoder Control Function (Control Method 1) at Down Count (Reverse Rotation)

(5) Flow Chart

The following figure shows the setting flow of this function.



The following figure shows the flow chart after interrupt processing.



The values of ENCA_n registers used in this function are as follows.

ENCA_nCTL[15:0] = 1000 0000 000x 01xx_B

ENCA_nIOC1[7:0] = 0000 00xx_B

ENCA_nCCR0 = (set any value)

ENCA_nCCR1 = (set any value)

ENCA_nCNT = (set any value)

“x” can be set arbitrarily. For the registers specifications, see **Section 28, Encoder Timer (ENCA)**.

The values of TSG3_n registers used in this function are as follows.

TSG3_nCTL0[7:0] = 000x 0011_B

TSG3_nCTL3[7:0] = 0000 00xx_B

TSG3_nCTL4[15:0] = 0000 0001 xxx0 0000_B

TSG3_nIOC0[7:0] = 0111 1110_B

TSG3_nIOC2[15:0] = 0xxx xxx0 0000 0000_B

TSG3_nOPT0[7:0] = 0011 1xx0_B

TSG3_nOPT1[7:0] = 0000 0xxx_B

TSG3_nCMP0 = (set any value)

TSG3_nCMP1W,5W,9W = (set any value)

TSG3_nCMP1,5,9 = (set any value)

TSG3_nPAT0W,1W = (set any value)

TSG3_nDTC0W,1W = (set any value)

“x” can be set arbitrarily. For the registers specifications, see **Section 25, Motor Control Timer (TSG3)**.

29.2.3.8 Two-Phase Encoder Control Function (Control Method 2)

(1) Overview

This function allows switching the phase lead and phase lag control of the motor control function (TSG3n) output patterns in 120-DC mode using the two-phase encoder control function (ENCAn).

(2) Configuration

The same configuration as the **Section 29.2.3.7, Two-Phase Encoder Control Function (Control Method 1)** applies to this function. See **(2) Configuration** of **Section 29.2.3.7, Two-Phase Encoder Control Function (Control Method 1)**.

(3) Registers

The same register values as the **Section 29.2.3.7, Two-Phase Encoder Control Function (Control Method 1)** applies to this function. See **(3) Registers** in **Section 29.2.3.7, Two-Phase Encoder Control Function (Control Method 1)**.

(4) Function

Details of this function are described using **Section 29.2.3.8, Two-Phase Encoder Control Function (Control Method 2)** with phase lead (normal rotation) as an example.

The following figure shows the timing diagram.

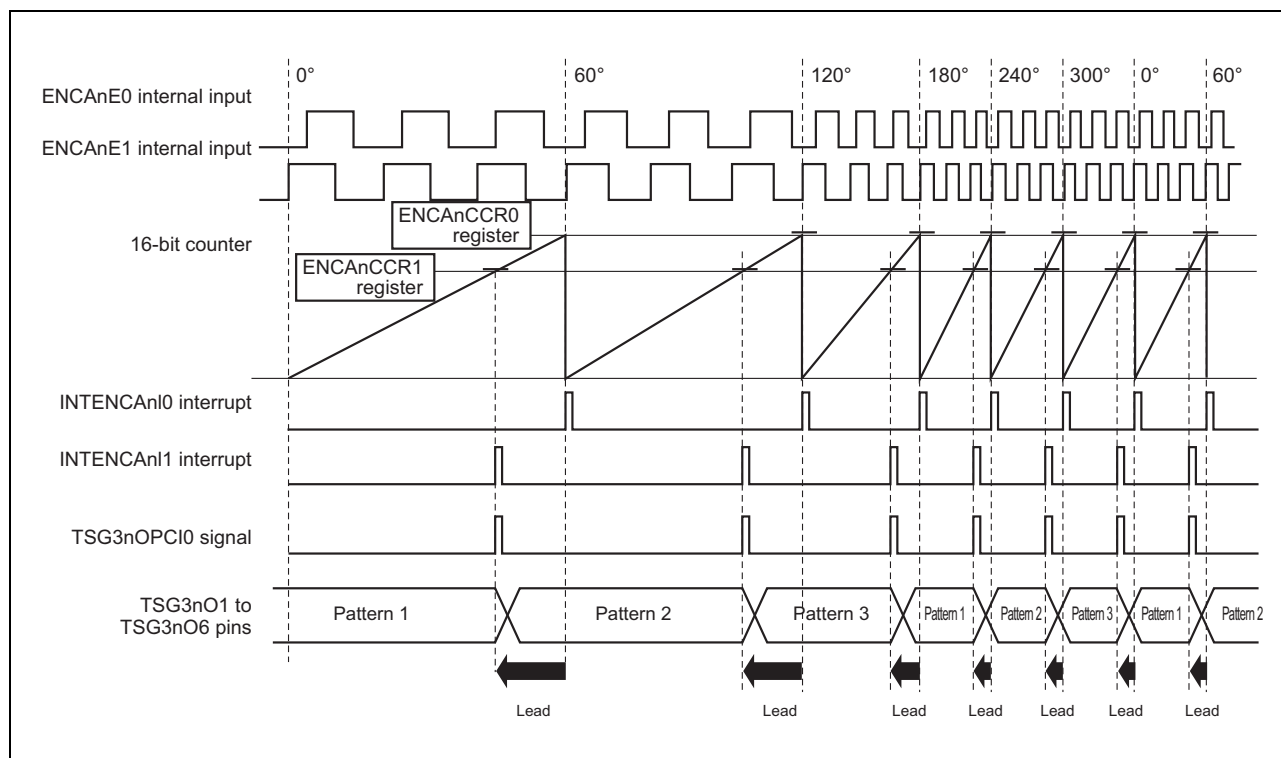


Figure 29.29 Timing Diagram of Two-Phase Encoder Control Function (Method 2) with Phase Lead (Normal Rotation)

- (1) Upon a match of the encoder counter value and ENCAnCCR1 (corresponds to the output pattern switch position for TSG3n), INTENCAnI1 is generated, and the set pattern is output from the TSG3nO1 to 6 pins.
- (2) Upon a match of the encoder counter value and ENCAnCCR0 (corresponds to the phase lead or lag of the switch position), INTENCAnI0 is generated and the encoder counter is cleared.

CAUTION

It is necessary to set ENCA_nCCR1 at each pattern switch (each INTENCA_n1 interrupt). It is necessary to match the initial output pattern of timer TSG3_n to the set value of ENCA_nCCR0 before starting, because this value is not cleared by the encoder clear input.

Switching between normal and reverse rotations of output patterns is set with the TSG3_nPSC bit of the TSG3_nOPT0 register.

The following figure shows the timing diagram with phase lag (normal rotation).

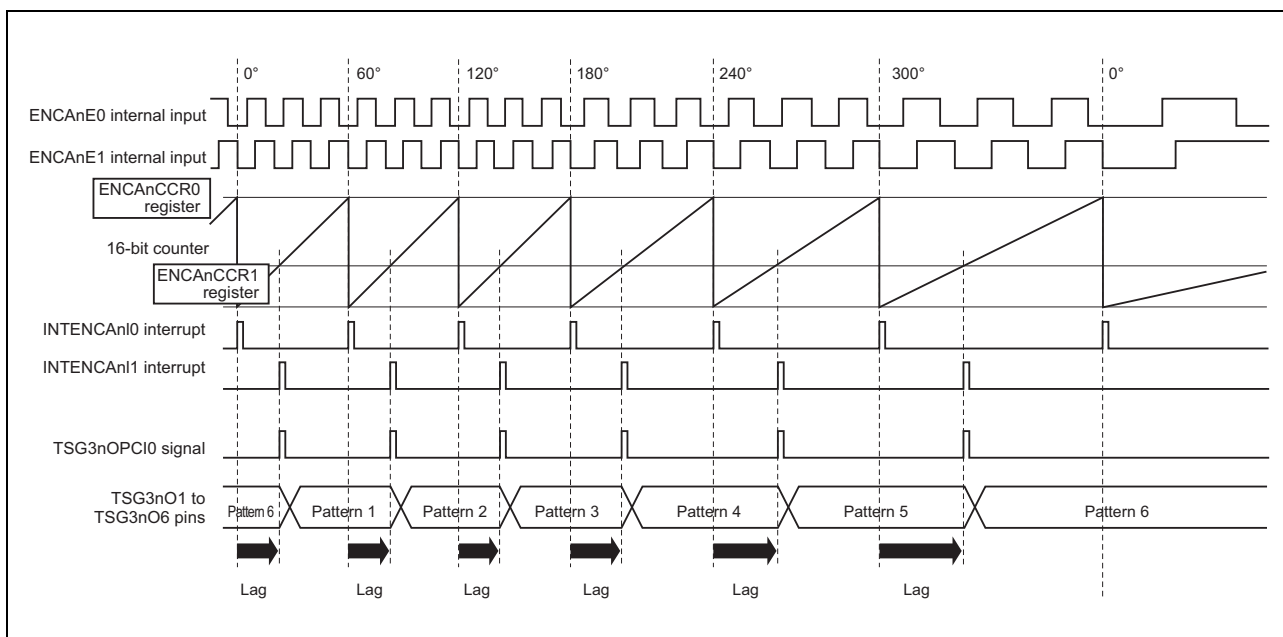


Figure 29.30 Timing Diagram of Two-Phase Encoder Control Function (Method 2) with Phase Lag (Normal Rotation)

By setting greater value to ENCA_nCCR1 than that of ENCA_nCCR0, TSG3_n output pattern phase can be made to lag.

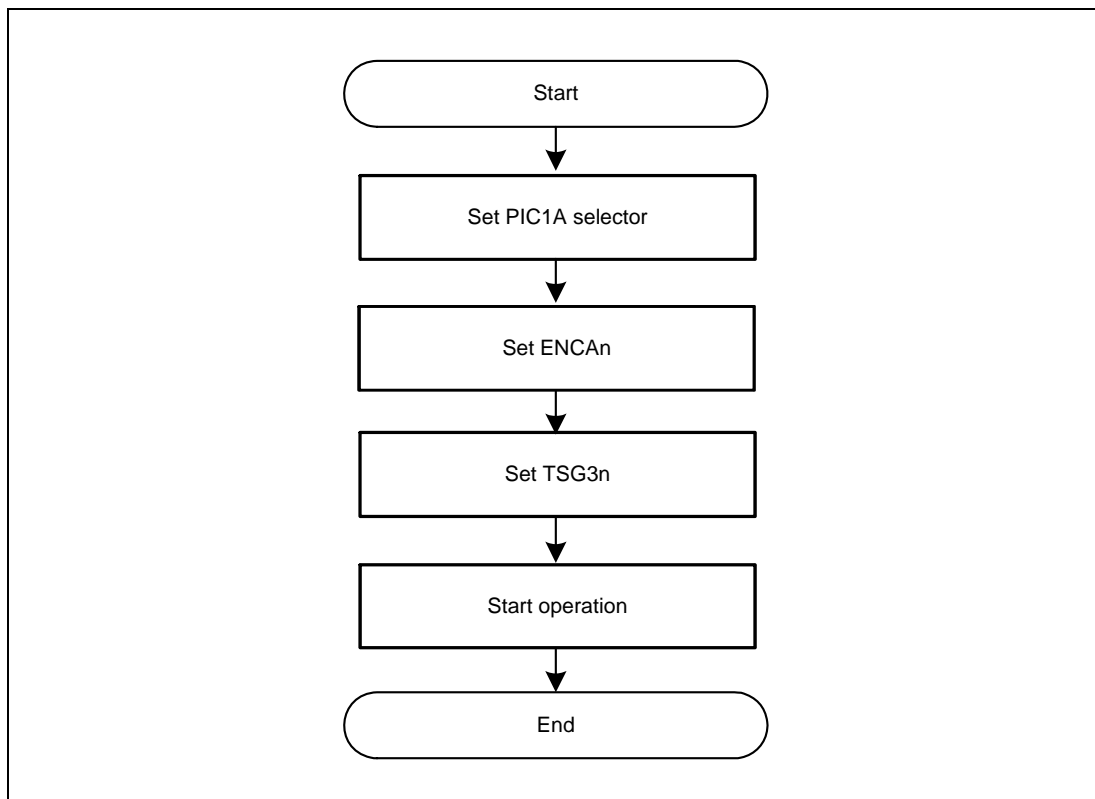
NOTE

This function can be used for phase lead and phase lag control both at up count and down count.

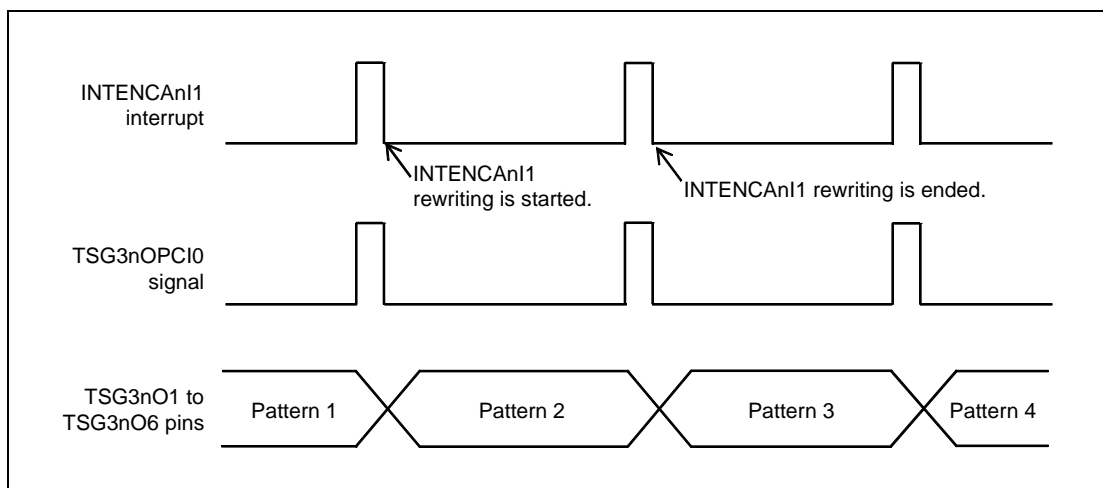
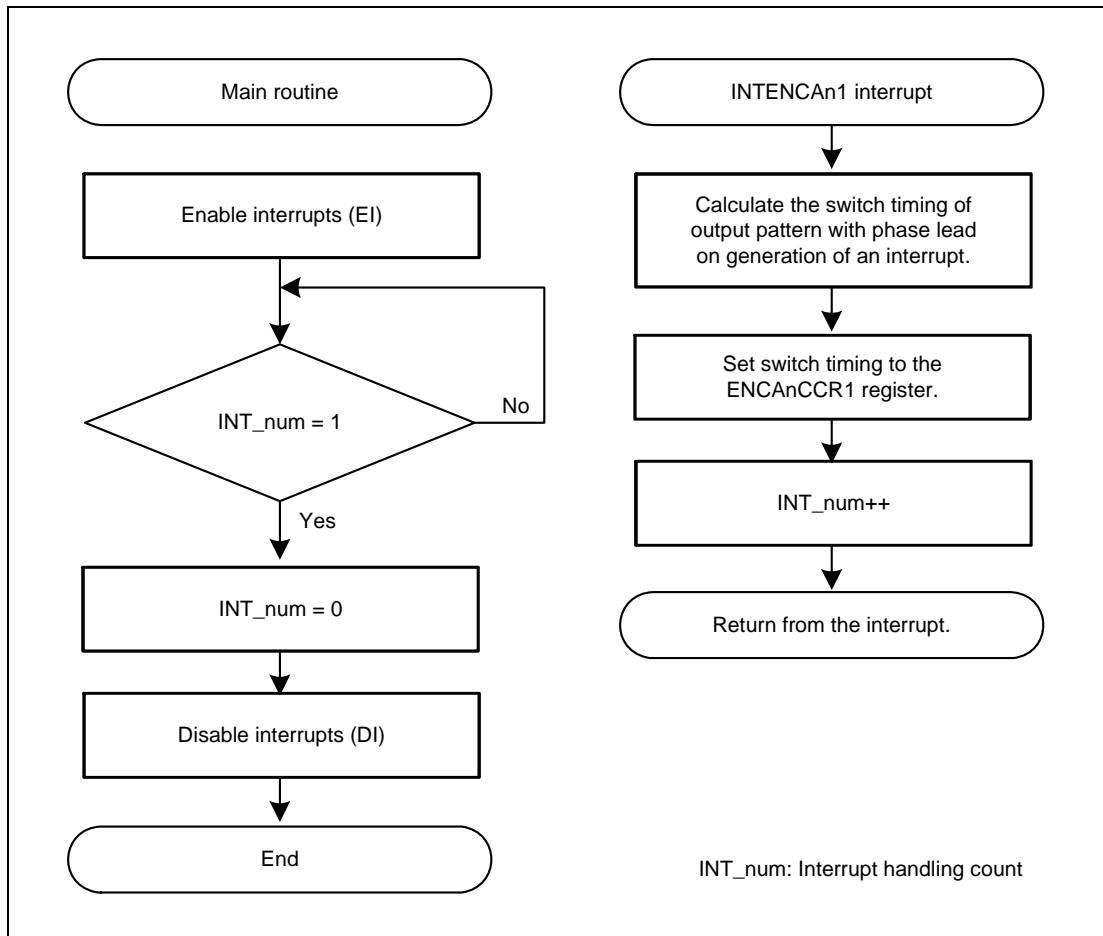
(5) Flow Chart

The flow charts for this function are shown below.

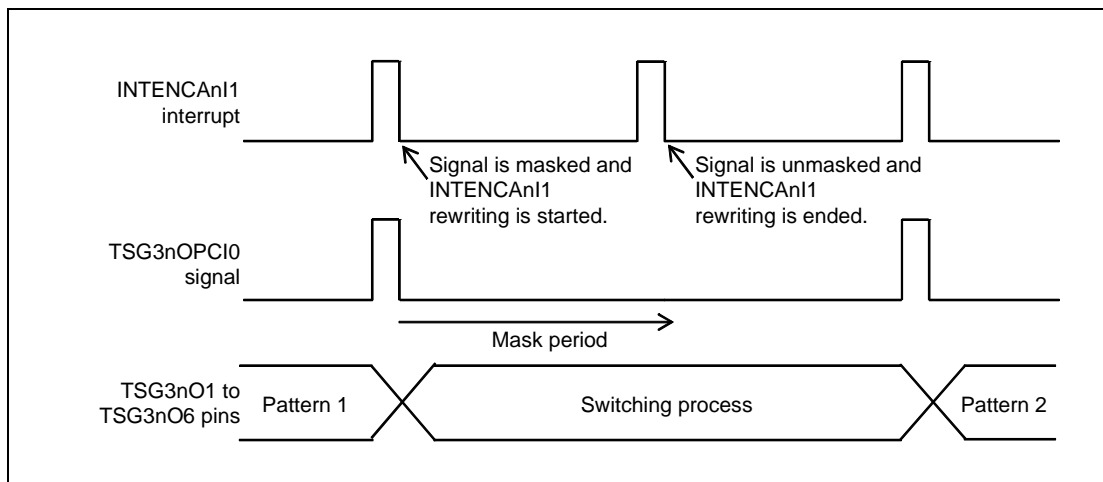
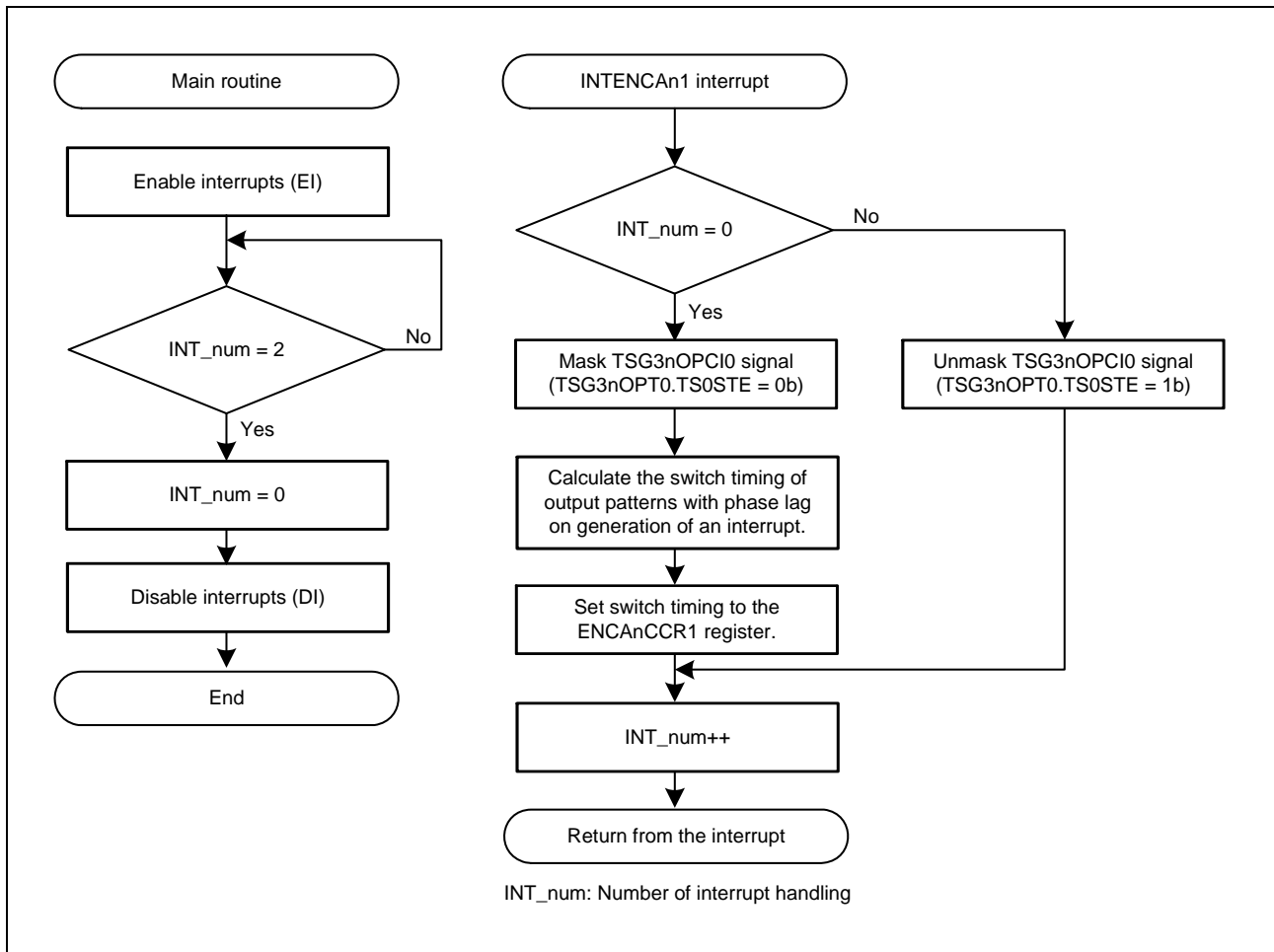
Flow chart of main routine.



Flow chart of ENCANCCR1 rewrite processing with phase lead.



Flow chart of ENCANCCR1 rewrite processing with phase lag.



The register settings for ENCA_n to use this function are as follows.

ENCA_nCTL[15:0] = 1000 0000 000x 01xx_B

ENCA_nIOC1[7:0] = 0000 00xx_B

ENCA_nCCR0 = (set any value)

ENCA_nCCR1 = (set any value)

ENCA_nCNT = (set any value)

“x” can be set arbitrarily. For the registers specifications, **Section 28, Encoder Timer (ENCA)**.

The register settings for TSG3_n to use this function are as follows.

TSG3_nCTL0[7:0] = 000x 0011_B

TSG3_nCTL3[7:0] = 0000 00xx_B

TSG3_nCTL4[15:0] = 0000 0001 xxx0 0000_B

TSG3_nIOC0[7:0] = 0111 1110_B

TSG3_nIOC2[15:0] = 0xxx xxx0 0000 0000_B

TSG3_nOPT0[7:0] = 0011 1xx0_B

TSG3_nOPT1[7:0] = 0000 0xxx_B

TSG3_nCMP0 = (set any value)

TSG3_nCMP1W, 5W, 9W = (set any value)

TSG3_nCMP1, 5, 9 = (set any value)

TSG3_nPAT0W, 1W = (set any value)

TSG3_nDTC0W, 1W = (set any value)

“x” can be set arbitrarily. For the registers specifications, **Section 25, Motor Control Timer (TSG3)**.

29.2.3.9 Three-Phase Pulse Input Control Function

(1) Overview

This function allows variable phase control of TSG3n pattern output in 120-DC mode using TSG3n and TAUDn.

The following diagram shows the method of three-phase pulse input control.

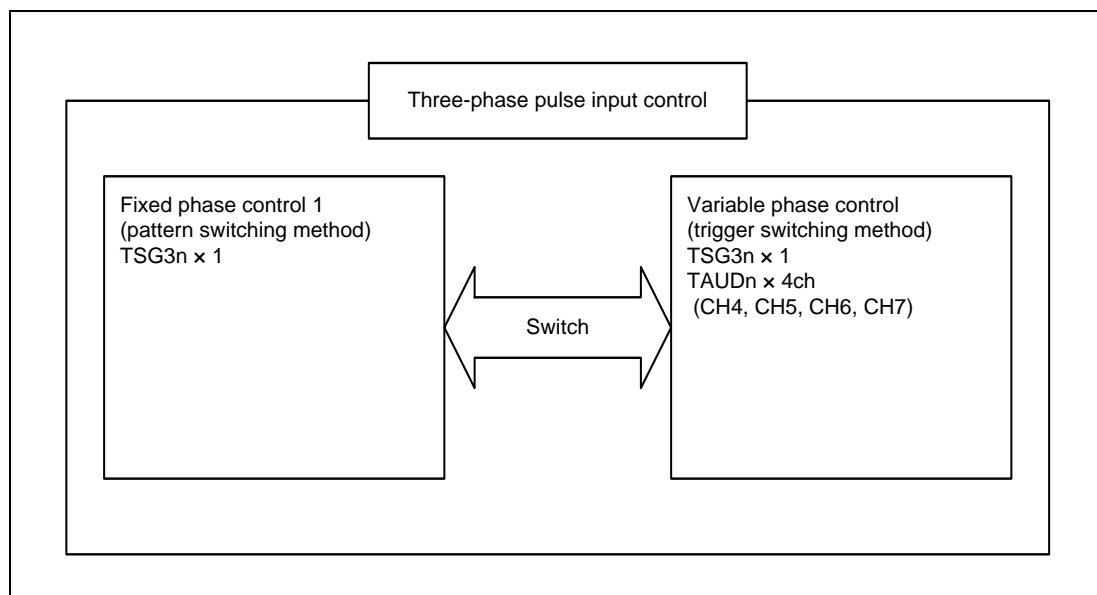


Figure 29.31 Control Method of Three-Phase Pulse Input Control Function

| Control Method | Function |
|---|--|
| Fixed phase control 1 (pattern switching method) | Outputs a fixed pattern at constant rotation angle. |
| Variable phase control (trigger switching method) | Varies the phase by arbitrary angle (or time) up to ± 60 degrees with reference to the rotation angle and outputs the pattern. |

(2) Configuration

Three-phase pulse input control function is realized by using three-phase pulse input and TAUDn offset trigger mode, and PIC1A in combination.

The following figure shows the block diagram of three-phase pulse input control function.

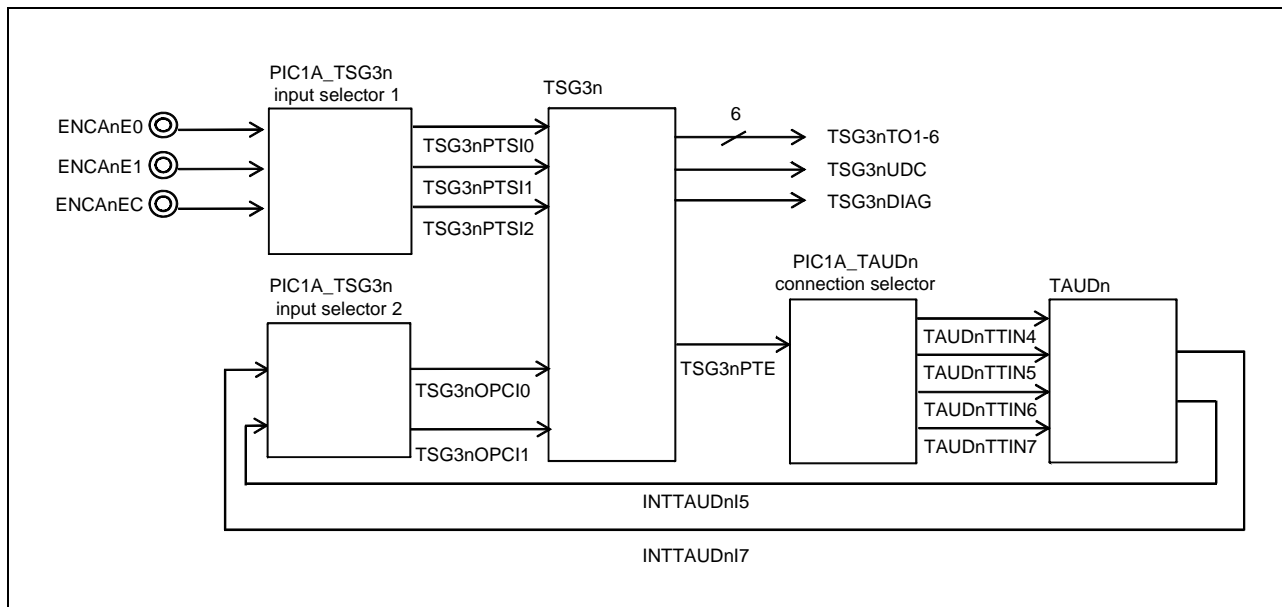


Figure 29.32 Block Diagram of Three-Phase Pulse Input Control

The configuration of this function is described below.

- [PIC1A_TSG3n input selector 1]
ENCA nE0, ENCA nE1, and ENCA nEC pin inputs are selected and output to TSG3nPTSIO to TSG3nPTSIO2.
- [TSG3n]
Patterns set in TSGnTO1 to TSGnTO6 are output in response to the input of TSG3nPTSIO to TSG3nPTSIO2 signals. TSG3nPTE is toggled each time the output patterns are switched.
- [PIC1A_TAUDn connection selector]
TSG3nPTE input is selected and output to TAUDnTTIN4 to TAUDnTTIN7.
- [TAUDn]
Interrupt signals INTTAUDnI5 and INTTAUDnI7 for output pattern phase generation are output with the offset trigger mode.
- [PIC1A_TSG3n input selector 2]
INTTAUDnI5 and INTTAUDnI7 inputs are selected and output as TSG3nOPCI0 and TSG3nOPCI1.

(3) Registers

Block diagram of PIC1A is shown in the following figure.

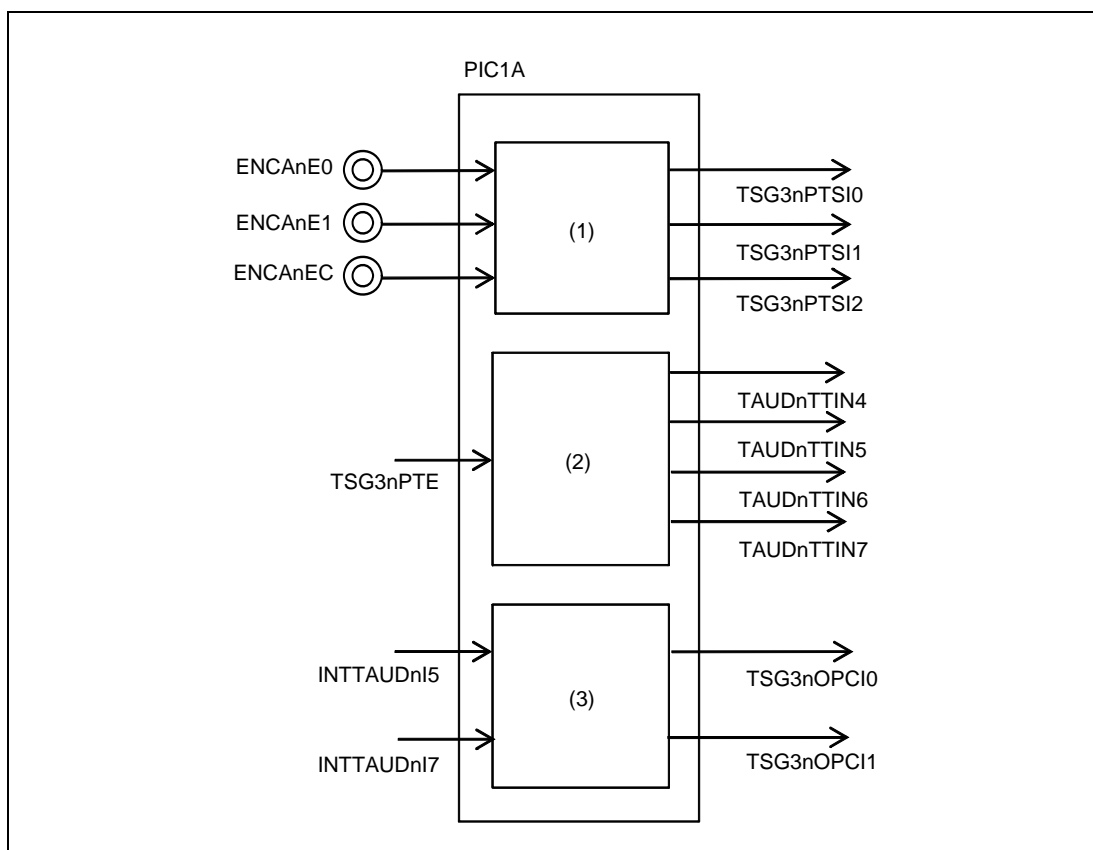


Figure 29.33 Block Diagram of PIC1A

The values of PIC1A registers used in this function are as follows.

Unit (1): PIC1A_TSG3n input selector 1

The values to output ENCA0E0, ENCA0E1, and ENCA0EC as TSG30PTSIO to TSG30PTS12

$$\text{PIC1ATSGHALLSEL}[0] = 1_{\text{B}}$$

$$\text{PIC1AREG50}[0] = 0_{\text{B}}$$

The values to output ENCA1E0, ENCA1E1, and ENCA1EC as TSG31PTSIO to TSG31PTS12

$$\text{PIC1ATSGHALLSEL}[1] = 1_{\text{B}}$$

$$\text{PIC1AREG51}[0] = 1_{\text{B}}$$

Unit (2): PIC1A_TAUDn connection selector

The values to output TSG3nPTE to TAUDnTTIN4 to TAUDnTTIN7.

$$\text{PIC1AREG2n0}[11:8] = 1010_{\text{B}}$$

$$\text{PIC1AREG2n0}[3:0] = 1111_{\text{B}}$$

$$\text{PIC1ATAUDnSEL}[15:8] = 00_{\text{H}}$$

Unit (3): PIC1A_ENCAn input selector 2

The values to output INTTAUDnI5 and INTTAUDnI7 to TSG3nOPCI0 and TSG3nOPCI1.

$$\text{PIC1AREG5n}[7:5] = 011_{\text{B}}$$

(4) Function

Details of the three-phase pulse input control function are described here.

The following figure shows the timing diagram.

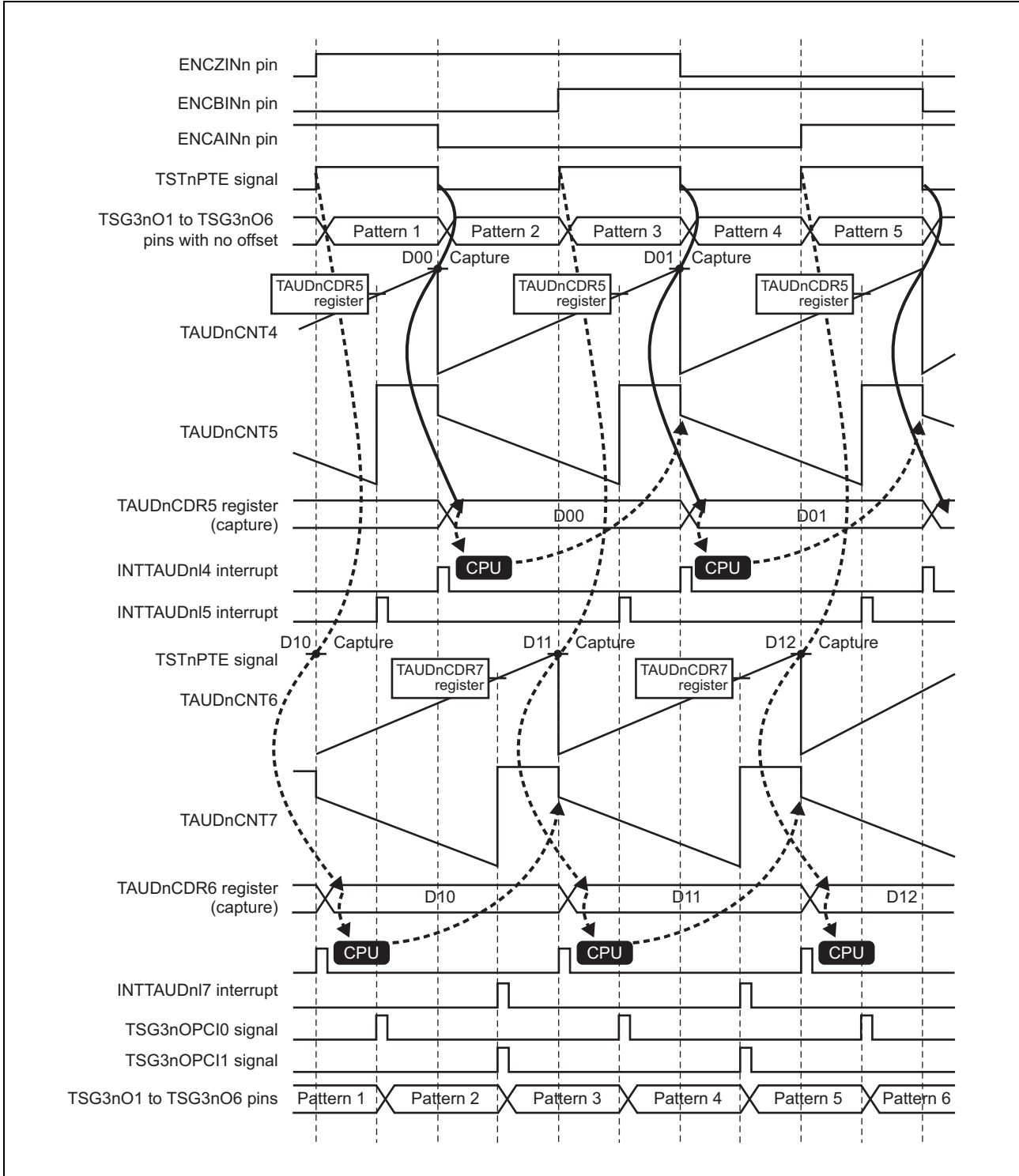


Figure 29.34 Three-Phase Pulse Input Control Function

- (1) The output patterns are switched by the TSG pattern switching method on detection of ENCA_n pin signal. The TSTnPTE signal toggles each time the patterns are switched.

- (2) TAUDnCNT6 and TAUDnCNT4 are captured at the rising edge and falling edge of the TSTnPTE signal. The interval to switch patterns is calculated according to the captured value.
- (3) CPU calculates the phase of the next output pattern and set the values to TAUDnCNT5 and TAUDnCNT7. The signals corresponding to the values are output as TSG3nOPCI0 and TSG3nOPCI1. At this time, the patterns delayed for the set phase are output by switching the output patterns by the trigger switching method.

The following table lists the relation between the value set to TAUDnCNT_m and the captured value in TAUDnCDR (m-1) (m = 5, 7).

| TAUDnCNT _m Register Setting | TSG3n Pattern Output Switch Timing |
|---|---|
| TAUDnCNT _m = 0000 _H | Patterns are switched upon detection of the TSTnPTE signal edge (delayed by up to one TAUDn count clock cycle). |
| TAUDnCNT _m = Captured value | Patterns are switched upon detection of the TSTnPTE signal edge. |
| TAUDnCNT _m < Captured value | The patterns are switched after a delay of one phase from detection of the TSTnPTE signal edge. |
| TAUDnCNT _m > Captured value | Setting prohibited |

An example of switch operation from fixed phase control 1 to variable phase control

The following figure shows an example of switch operation from fixed phase control 1 to variable phase control.

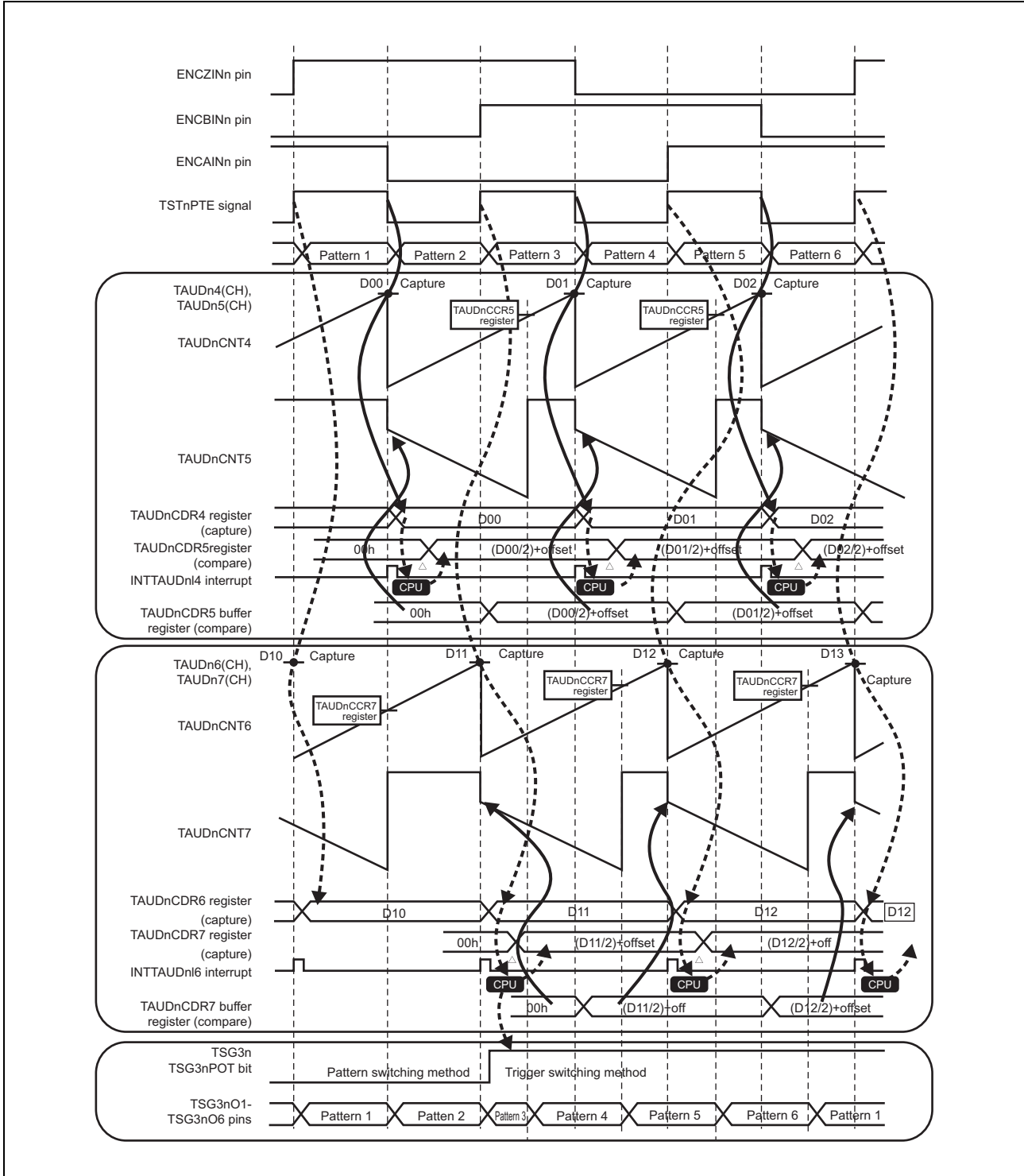


Figure 29.35 An Example of Switch Operation from Fixed Phase Control 1 to Variable Phase Control

By changing the TSG3nPOT bit from low level to high level, the output pattern switching method is changed to the trigger switching method, and the variable phase control is enabled.

An example of switch operation from variable phase control to fixed phase control 1

The following figure shows an example of switch operation from variable phase control to fixed phase control 1.

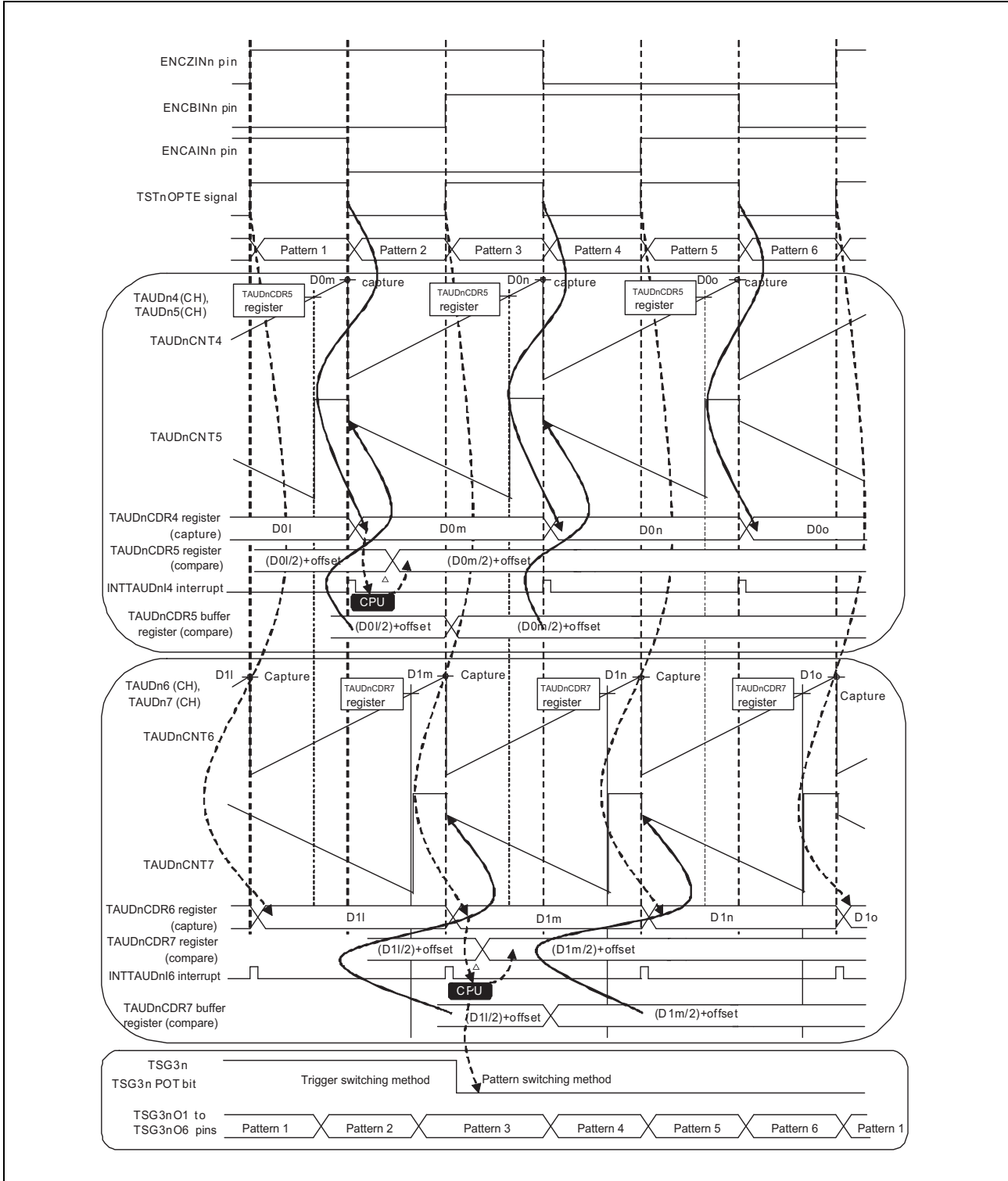


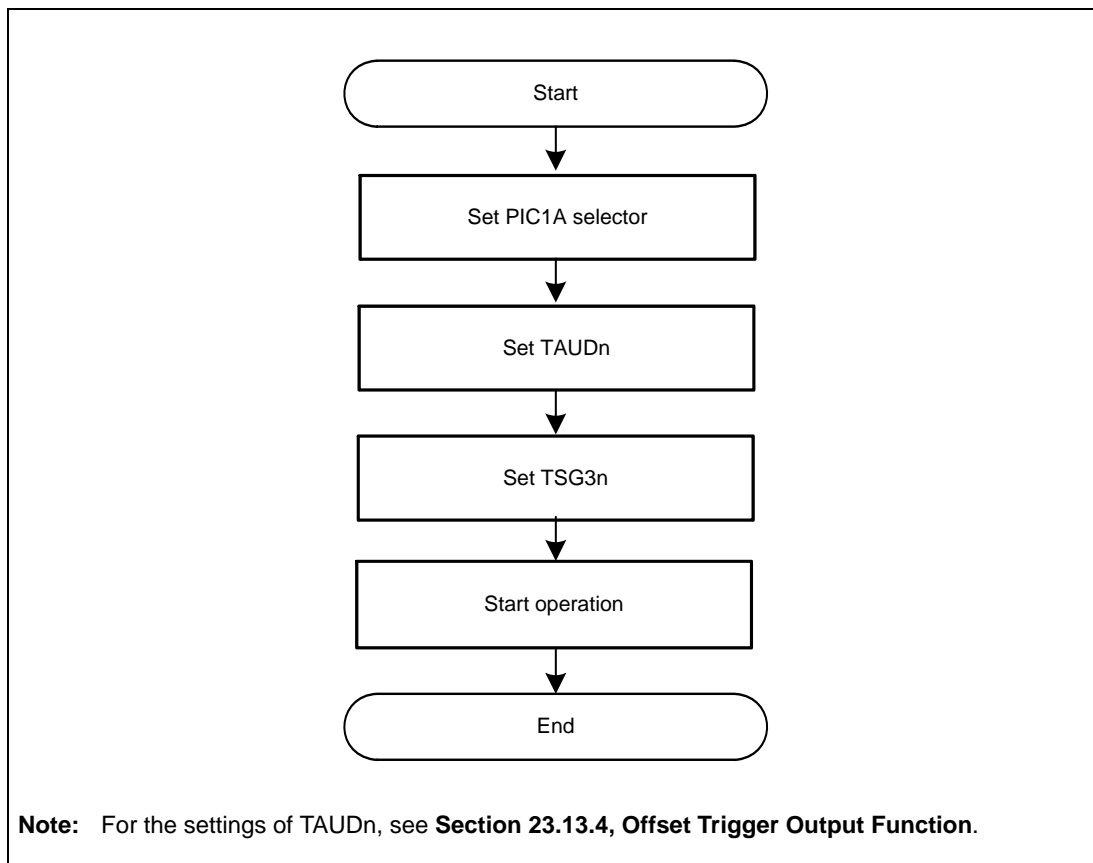
Figure 29.36 Example of Switch Operation from Variable Phase Control to Fixed Phase Control 1

By changing the TSG3nPOT bit from high level to low level, the output pattern switching method is changed to the pattern switching method, and the fixed phase control 1 is enabled.

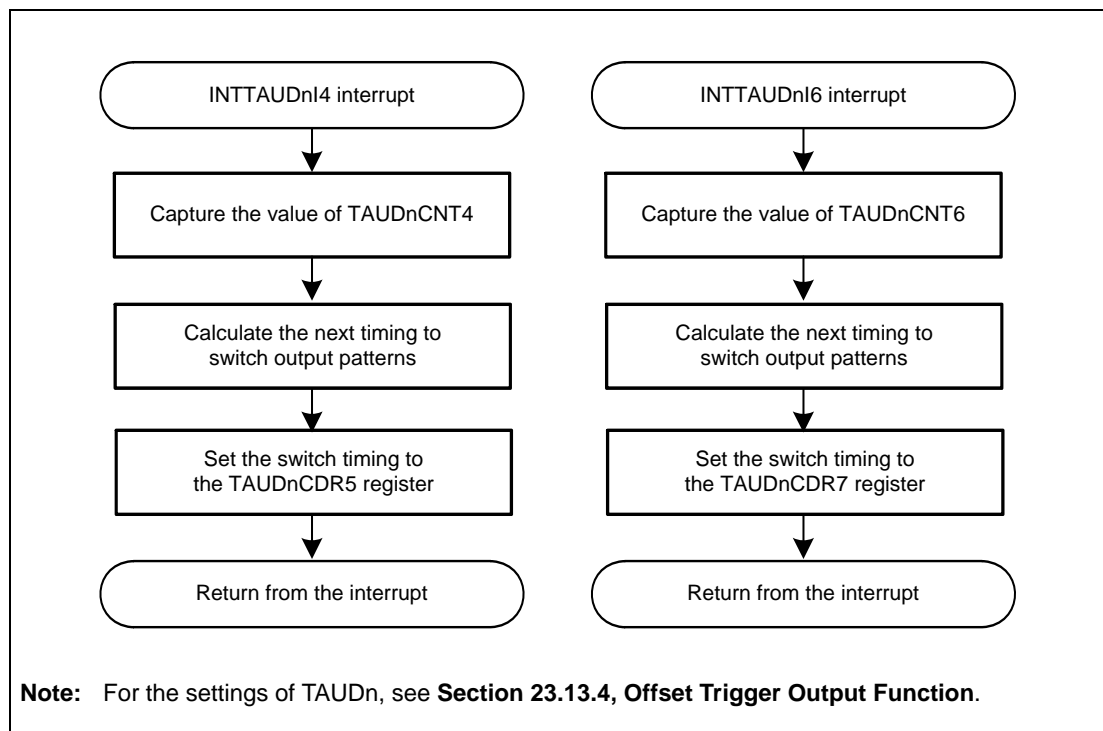
(5) Flow Chart

The flow charts for this function are shown below.

Flow chart of main routine



Flow chart for rewriting of TAUDnCDR5 and TAUDnCDR7 during operation



The values of TSG3n registers used in this function are as follows.

$TSG3nCTL0[7:0] = 0000\ 0011_B$

$TSG3nCTL3[7:0] = 0000\ 00xx_B$

$TSG3nCTL4[15:0] = 0000\ 0001\ xxx0\ 0000_B$

$TSG3nIOC0[7:0] = 0111\ 1110_B$

$TSG3nIOC1[7:0] = 0001\ xxxx_B$

$TSG3nIOC2[15:0] = 0xxx\ xxx0\ 0000\ 0000_B$

$TSG3nOPT0[7:0] = 0011\ 1xx0_B$

$TSG3nOPT1[7:0] = 0000\ 0xxx_B$

$TSG3nCMP0 = (\text{set any value})$

$TSG3nCMP1W,5W,9W = (\text{set any value})$

$TSG3nCMP1,5,9 = (\text{set any value})$

$TSG3nPAT0W,1W = (\text{set any value})$

$TSG3nDTC0W,1W = (\text{set any value})$

“x” can be set arbitrarily. For the registers specifications, see **Section 25, Motor Control Timer (TSG3)**.

29.2.3.10 Three-Phase Encoder Control Function

(1) Overview

The function allows three-phase external pattern inputs (TSG3nPTSIO to TSG3nPTS12) to be encoded using ENCA_n.

(2) Configuration

The three-phase encoder control function is realized by using TSG3_n, ENCA_n, and PIC1A in combination. The following figure shows the block diagram of three-phase encoder control function.

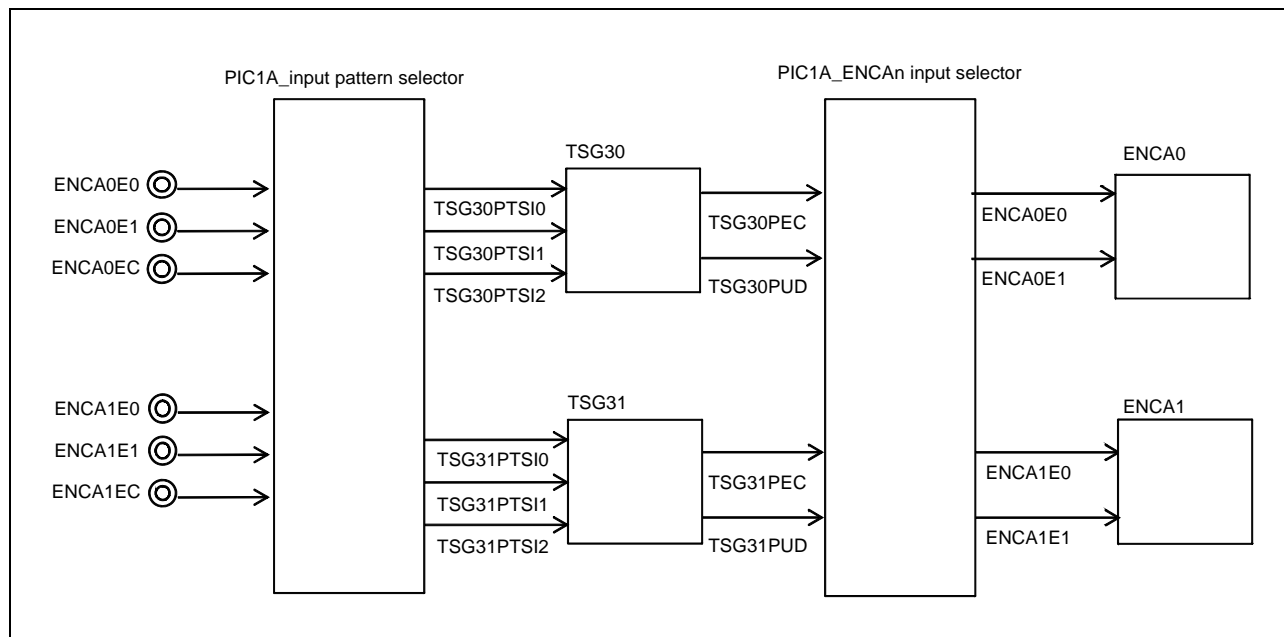


Figure 29.37 Block Diagram of Three-Phase Encoder Control Function

The configuration of this function is described below.

- [PIC1A_input pattern selector]
ENCA_nE0, ENCA_nE1, and ENCA_nEC pin inputs are selected and output to TSG3_nPTSIO to TSG3_nPTS12.
- [TSG3_n]
Patterns set in TSG3_nPEC are output in response to the input of the TSG3_nPTSIO to TSG3_nPTS12 signals. TSG3_nPUD is output depending on the rotation (normal or reverse).
- [PIC1A_ENCA_n input selector]
TSG3_nPEC is selected and output to ENCA_nE0. TSG3_nPUD is selected and output to ENCA_nE1.
- [ENCA_n]
ENCA_nE0 and ENCA_nE1 are encoded.

(3) Registers

Block diagram of PIC1A is shown in the following figure.

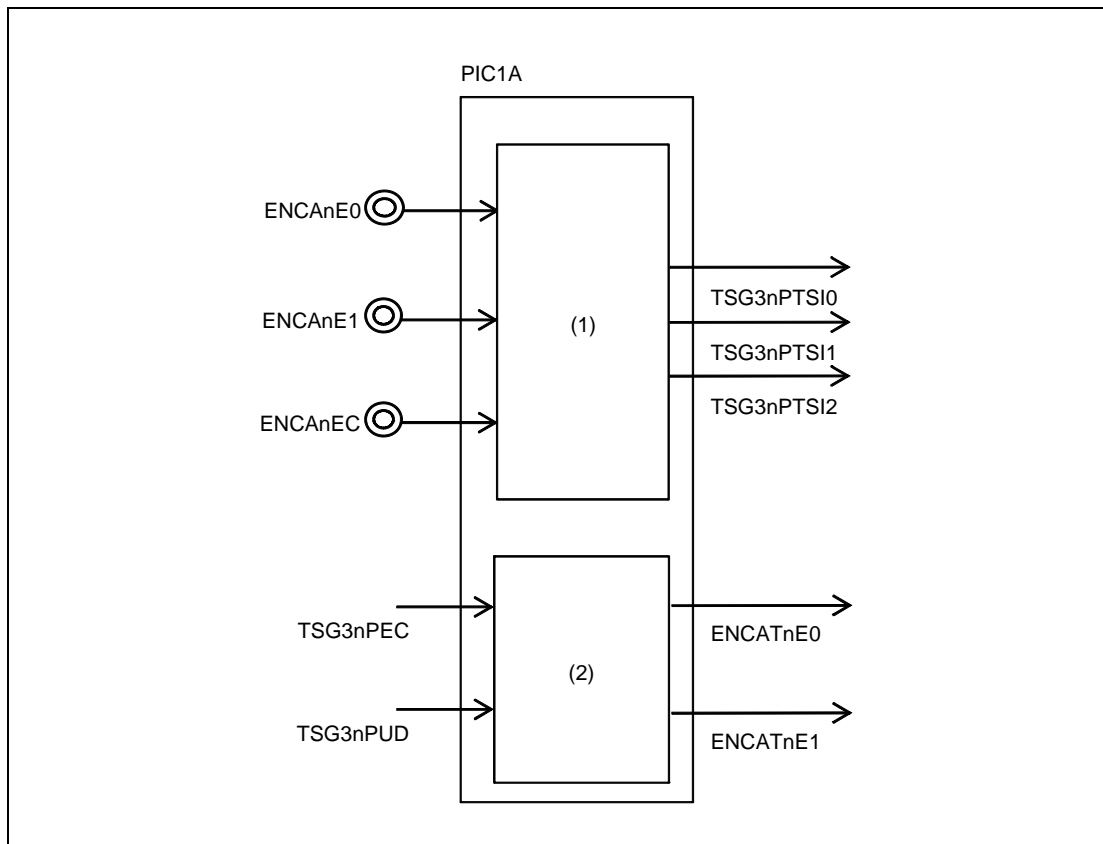


Figure 29.38 Block Diagram of PIC1A

The register settings for PIC1A to use this function are as follows.

Unit (1): PIC1A_ input pattern selector

The values to output ENCA0E0, ENCA0E1, and ENCA0EC as TSG30PTSIO to TSG30PTS2

$$\text{PIC1ATSGHALLSEL}[0] = 1_{\text{B}}$$

$$\text{PIC1AREG50}[0] = 0_{\text{B}}$$

The value to output ENCA1E0, ENCA1E1, and ENCA1EC as TSG31PTSIO to TSG31PTS2

$$\text{PIC1ATSGHALLSEL}[1] = 1_{\text{B}}$$

$$\text{PIC1AREG51}[0] = 1_{\text{B}}$$

Unit (2): PIC1A_ENCA_n input selector

The values to output the TS0PEC and TS0PUD as ENCA0E0 and ENCA0E1, respectively

$$\text{PIC1AREG30}[22] = 0_{\text{B}}$$

$$\text{PIC1AREG30}[1:0] = 11_{\text{B}}$$

The values to output the TS1PEC and TS1PUD as ENCA1E0 and ENCA1E1, respectively

$$\text{PIC1AREG30}[9:8] = 10_{\text{B}}$$

$$\text{PIC1AREG30}[7:6] = 10_{\text{B}}$$

(4) Function

Details of the three-phase encoder control function are described below.

The following figure shows the timing diagram.

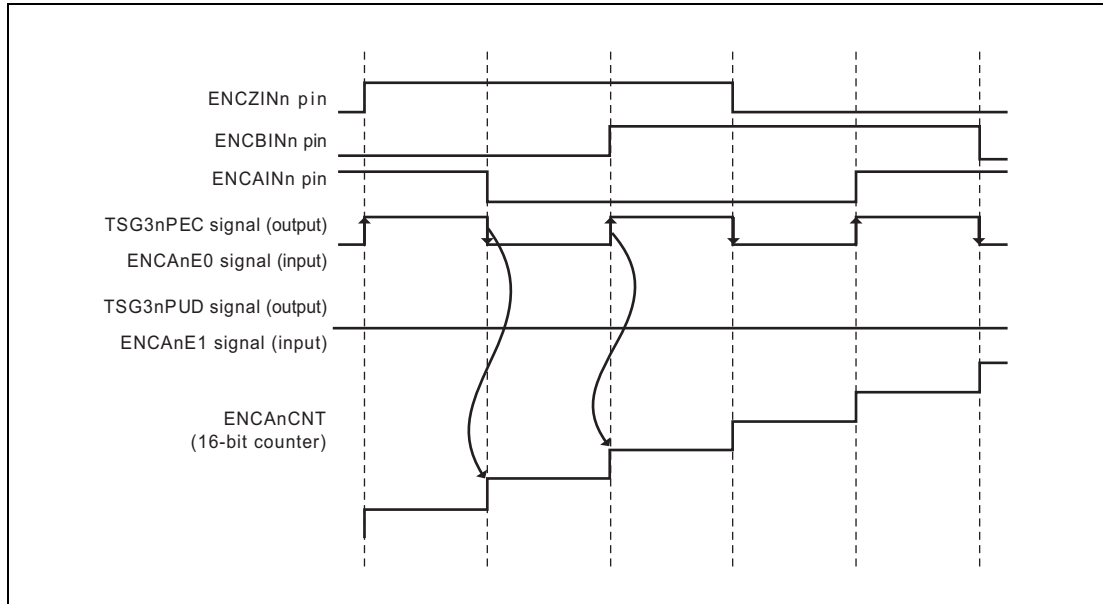


Figure 29.39 Timing Diagram of Three-Phase Encoder Control Function_ENCAnUDS1 and ENCAAnUDS0 = 00_B

- (1) When the low level is input to ENCAAnE1, the count is incremented each time an active edge is input to ENCAAnE0.

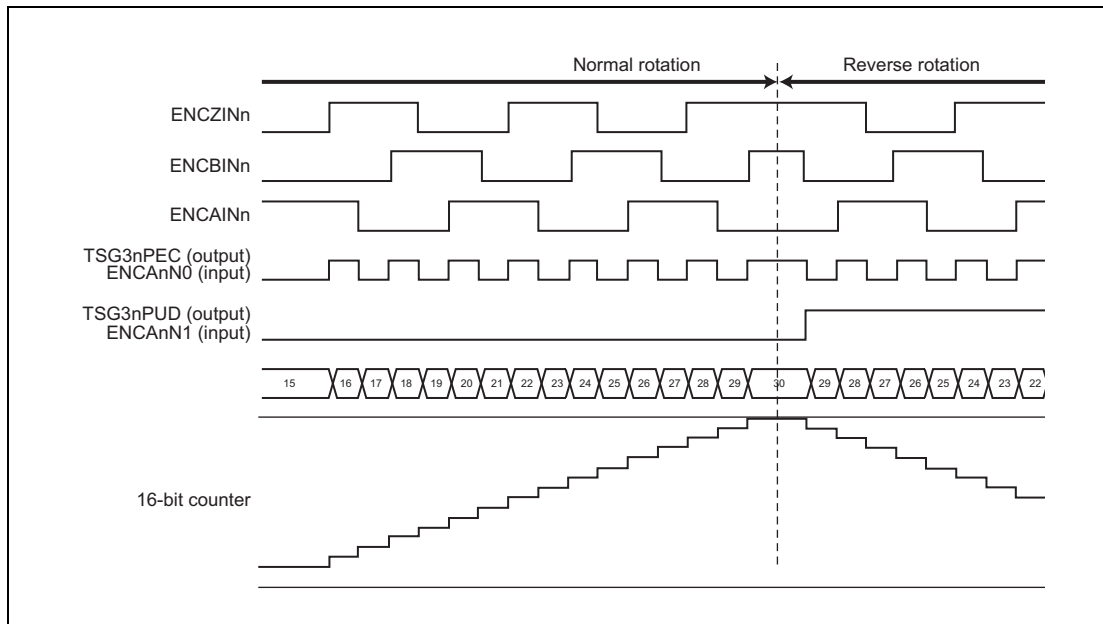


Figure 29.40 Timing Diagram of Three-Phase Encoder Control Function_Normal Rotation/Reverse Rotation

(5) Flow Chart

Set PIC before using the three-phase encoder control function.

The values of ENCA_n registers used in this function are as follows.

$$\text{ENCA}_n\text{CTL}[15:0] = \text{xx}00\ 00\text{xx}\ 000\text{x}\ \text{xx}00_{\text{B}}$$

$$\text{ENCA}_n\text{IOC1}[7:0] = 0000\ 00\text{xx}_{\text{B}}^{*1}$$

$$\text{ENCA}_n\text{CCR0} = (\text{set any value})$$

$$\text{ENCA}_n\text{CCR1} = (\text{set any value})$$

$$\text{ENCA}_n\text{CNT} = (\text{set any value})$$

“x” can be set arbitrarily. For the registers specifications, **Section 28, Encoder Timer (ENCA)**.

Note 1. Except for ENCA_nIOC1[1:0] = 00_B (no edge detected) because edge detection is necessary.

The values of TSG3_n registers used in this function are as follows.

$$\text{TSG3}_n\text{CTL0}[7:0] = 0000\ 0001_{\text{B}}$$

$$\text{TSG3}_n\text{CTL3}[7:0] = 0000\ 00\text{xx}_{\text{B}}$$

$$\text{TSG3}_n\text{CTL4}[15:0] = 0000\ 000\text{x}\ \text{xxxx}\ \text{xxxx}_{\text{B}}$$

$$\text{TSG3}_n\text{IOC0}[7:0] = 0\text{xxx}\ \text{xxx}0_{\text{B}}$$

$$\text{TSG3}_n\text{IOC1}[7:0] = 0001\ \text{xxxx}_{\text{B}}$$

$$\text{TSG3}_n\text{IOC2}[15:0] = 0\text{xxx}\ \text{xxx}0\ 0000\ 0000_{\text{B}}$$

$$\text{TSG3}_n\text{OPT0}[7:0] = 0\text{xxx}\ \text{xxx}0_{\text{B}}$$

$$\text{TSG3}_n\text{OPT1}[7:0] = 0000\ 0\text{xxx}_{\text{B}}$$

$$\text{TSG3}_n\text{CMP0} = (\text{set any value})$$

$$\text{TSG3}_n\text{CMP1W,5W,9W} = (\text{set any value})$$

$$\text{TSG3}_n\text{CMP1,5,9} = (\text{set any value})$$

$$\text{TSG3}_n\text{PAT0W,1W} = (\text{set any value})$$

$$\text{TSG3}_n\text{DTC0W,1W} = (\text{set any value})$$

“x” can be set arbitrarily. For the registers specifications, **Section 25, Motor Control Timer (TSG3)**.

29.2.3.11 TAUD Input Select Function

(1) Overview

The function selects TAUDn input signal to be input as TAUDnTINm/m+1 signal from either TAUD0TINm/m+1 signal or TAUD1TINm/m+1 signal (m is an even number between 0 and 15).

(2) Configuration

The TAUD input select function is realized by using TAUDn input signals and PIC1A in combination.

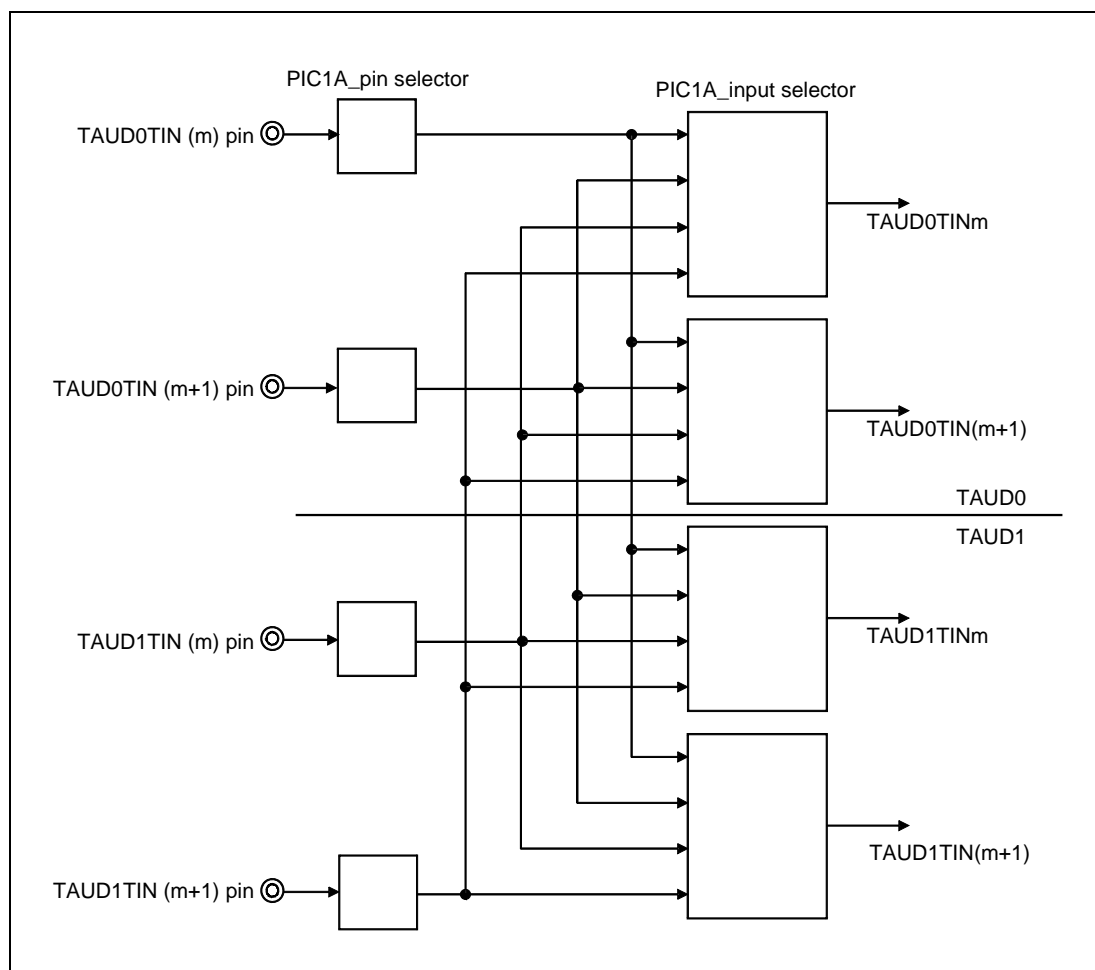


Figure 29.41 Block Diagram of TAUD Input Select Function

(3) Registers

The register settings for PIC1A to use this function are as follows.

PIC1A_pin selector

Select TAUDn external channel input pin for output from PIC1A pin selector.

PIC1A_input selector

Select the input signal using the following registers.

PIC1ATAUD0SEL

PIC1ATAUD1SEL

For the details of the register settings, **Section 29.2.2.10**, **Section 29.2.2.11**, **Section 29.2.2.18**, **Section 29.2.2.19**, **Section 29.2.2.21**, and **Section 29.2.2.24** .

(4) Function

Details of this function are described using selection of the TAUD0TIN[1:0] signal as an example.

The following table lists the selection of the TAUD0TIN[1:0] signals as an example. Setting 000000 to PIC1AREG31[11:6], and setting 01 to PIC1ATAUD0SEL[3:2] and PIC1ATAUD0SEL[1:0] causes the TAUD0TIN0 and TAUD0TIN1 signals to be input to TAUD0TIN1 and TAUD0TIN0 input pins of the TAUD0 timer. Setting 1 to PIC1ATAUD0SEL[3] and PIC1ATAUD0SEL[1] selects TIN pin signal of TAUD1.

| Register Setting | |
|------------------|---------------|
| PIC1ATAUD0SEL | |
| [1:0] | TAUD0TIN0 |
| 00 _B | TAUD0TIN0 pin |
| 01 _B | TAUD0TIN1 pin |
| 10 _B | TAUD1TIN0 pin |
| 11 _B | TAUD1TIN1 pin |

| Register Setting | |
|------------------|---------------|
| PIC1ATAUD0SEL | |
| [3:2] | TAUD0TIN1 |
| 00 _B | TAUD0TIN1 pin |
| 01 _B | TAUD0TIN0 pin |
| 10 _B | TAUD1TIN1 pin |
| 11 _B | TAUD1TIN0 pin |

(5) Flow Chart

Set PIC1A before starting the TAUDn timer.

29.2.3.12 Hi-Z Control Function

(1) Overview

The function disconnects three-phase output signal and changes to Hi-Z state.

For details of the purpose and operation of the Hi-Z control function, see **Section 26.4.1, Asynchronous Hi-Z Control Function.**

(2) Configuration

The ESO_n, ERROROUT, INTTSG30IER, INTTSG31IER, INTADCG0ERR, and INTADCG1ERR signals are masked and OR'ed in PIC1A, and output to TAPAn as the signal for Hi-Z control.

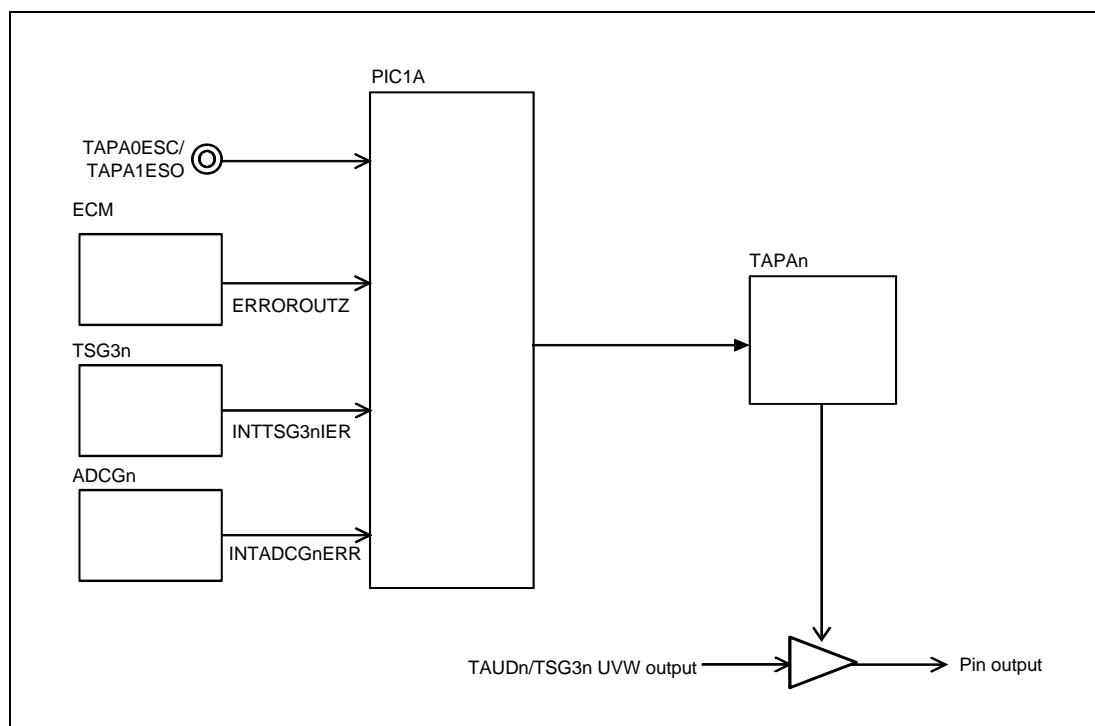


Figure 29.42 Block Diagram of Hi-Z Control

(3) Registers

The block diagram of PIC1A is shown in the following figure.

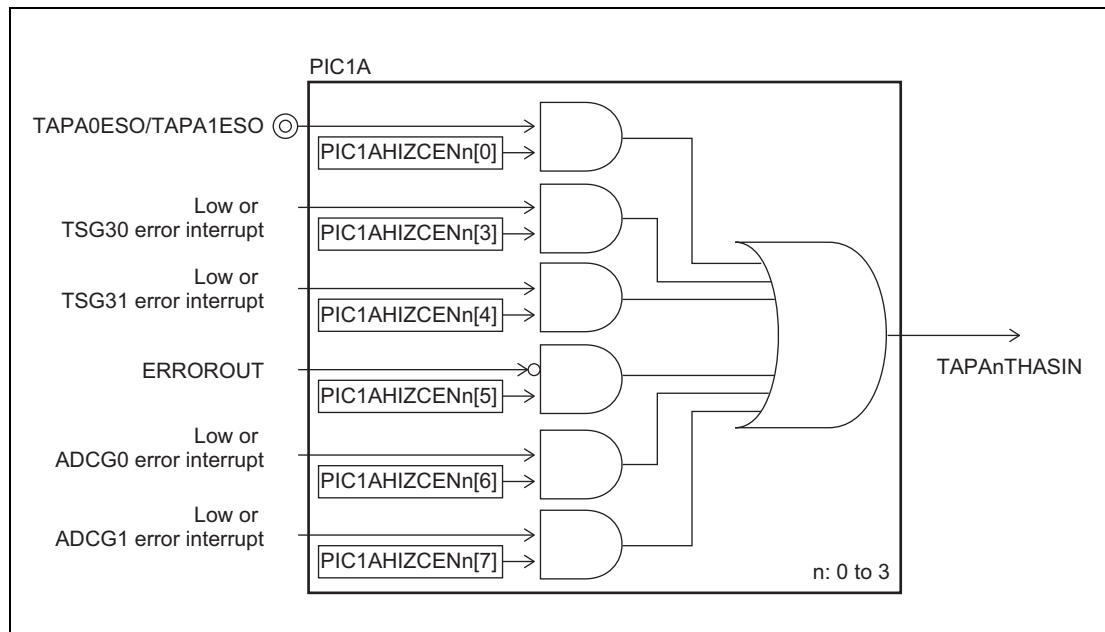


Figure 29.43 Block Diagram of PIC1A

The register settings for PIC1A to use this function are as follows.

PIC1AHIZCENn[5] = 1 (enabled), 0 (disabled): ERROROUT

PIC1AHIZCENn[0] = 1 (enabled), 0 (disabled): TAPA0ESO or TAPA1ESO

PIC1AHIZCENn[3] = 1 (enabled), 0 (disabled): INTTSG30IER

PIC1AHIZCENn[4] = 1 (enabled), 0 (disabled): INTTSG31IER

PIC1AHIZCENn[6] = 1 (enabled), 0 (disabled): INTADCG0ERR

PIC1AHIZCENn[7] = 1 (enabled), 0 (disabled): INTADCG1ERR

n = 0, 1 is available for TAUD0/TAUD1 and not available for PIC1AHIZCENn[3] and PIC1AHIZCENn[4].

n = 2 is available for TSG30 and not available for PIC1AHIZCENn[4].

n = 3 is available for TSG31 and not available for PIC1AHIZCENn[3].

Table 29.41 Correspondence between the Hi-Z Control Function Input Pins and Control Pins

| Input Pin Name | Control Register | TAPA Unit Number | Target Timers for Hi-Z Control |
|----------------|------------------|------------------|--------------------------------|
| TAPA0ESO | PIC1AHIZCEN0 | TAPA0 | TAUD0 |
| TAPA1ESO | PIC1AHIZCEN1 | TAPA1 | TAUD1 |
| TAPA0ESO | PIC1AHIZCEN2 | TAPA2 | TSG30 |
| TAPA1ESO | PIC1AHIZCEN3 | TAPA3 | TSG31 |

(4) Function

The TAPA0ESO/TAPA1ESO pins, ERROROUT, INTTSG30IER, INTTSG31IER, INTADCG0ERR, and INTADCG1ERR for Hi-Z control are masked and OR'ed in PIC1A, and output to TAPAn. For Hi-Z control by TAPA, see **Section 26.4.1.3, Basic Operation**.

(5) Flow Chart

Set PIC1A before starting Hi-Z control.

For the operation flow of TAPA, see **Section 26.4.1, Asynchronous Hi-Z Control Function**.

29.2.3.13 Timer Output Monitor Function (PWM-Diag)

(1) Overview

With this function, output signals of TAUD0, TAUD1, TSG30, TSG31, OSTM0, and OSTM1 are monitored using TAUD2.

(2) Configuration

Monitoring of output signals of TAUD0, TAUD1, TSG30, TSG31, OSTM0, and OSTM1 is realized using TAUD2 and PIC1A in combination.

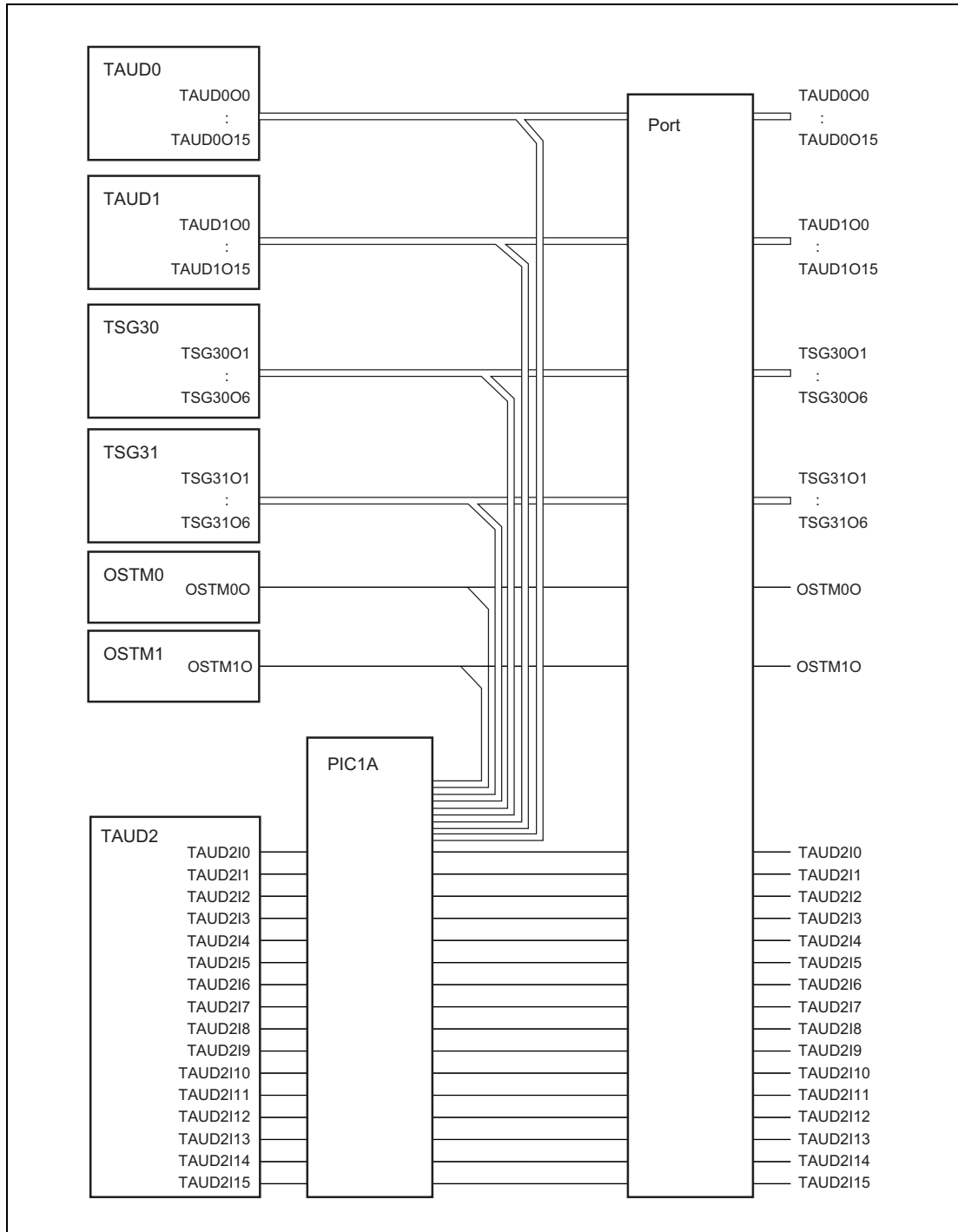


Figure 29.44 Block Diagram of Timer Output Monitor Function

(3) Registers

- The POMONSEL register selects the output signal to be monitored.
- Set the alternative output level loopback function for the port of the output signal to be monitored.
For details, see **Section 2.3.4.3, (1) PBDCn/JPBDC0 — Port Bi-Direction Control Register**.

29.2.3.14 Timer Input Monitor Function**(1) Overview**

With this function, input signals of TAUJ0 and TAUJ1 are monitored using TAUJ2.

(2) Configuration

Monitoring of input signals of TAUJ0 and TAUJ1 is realized using TAUJ2 and PIC1A in combination.

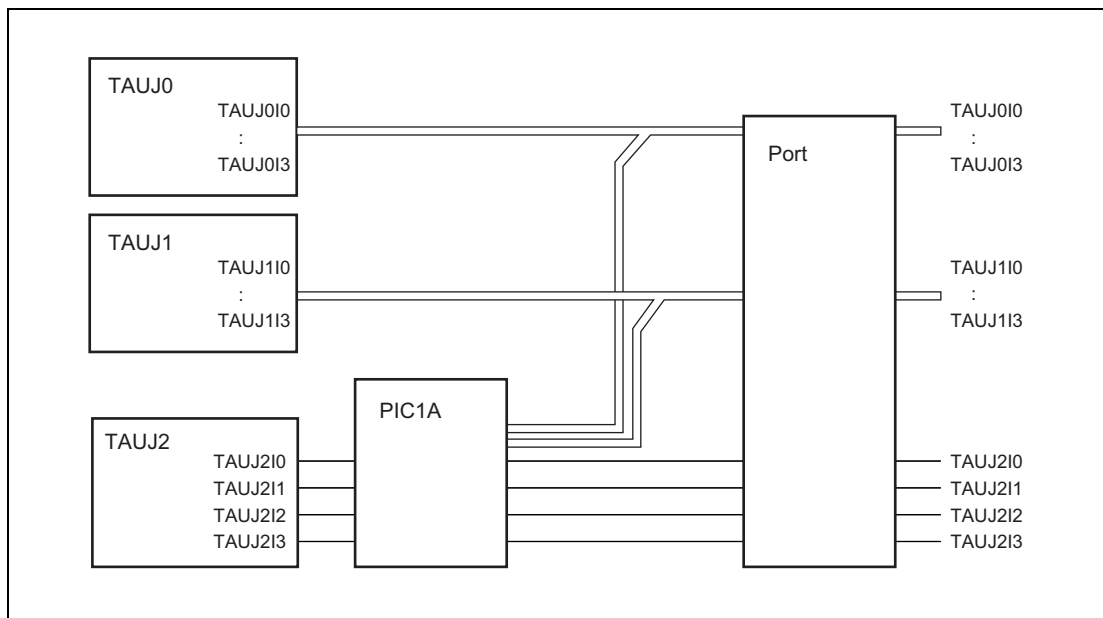


Figure 29.45 Block Diagram of Timer Input Monitor Function

(3) Registers

The PIMONSEL register selects the input signals to be monitored.

29.2.3.15 TSG3 Synchronous Clear Function

(1) Overview

This function allows synchronous starting and clearing of TAUD0, TSG30 and TSG31.

(2) Configuration

The interrupt signals from TAUD0 channels 14 and 15 are used as synchronous start triggers to initiate synchronous starting or clearing of TSG30 and TSG31.

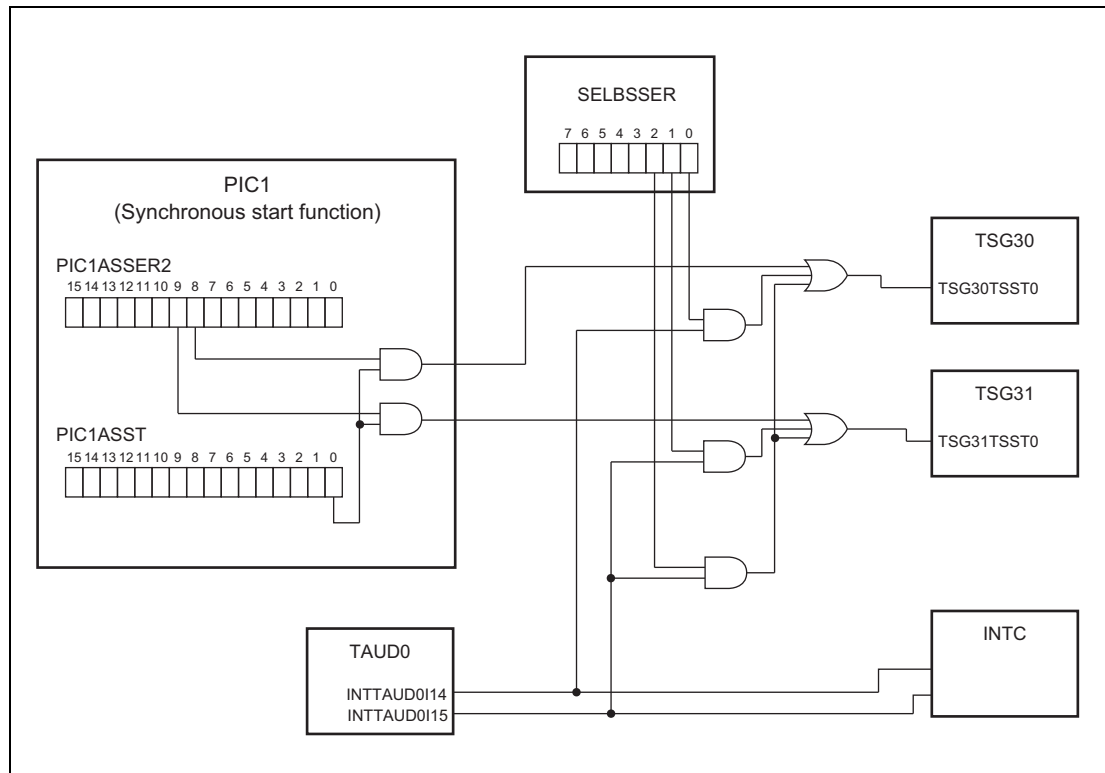


Figure 29.46 Block Diagram of TSG3 Synchronous Clear Function

29.3 Peripheral Interconnection — 2 (PIC2B)

29.3.1 Overview

29.3.1.1 Functional Overview

The peripheral interconnection-2 (PIC2B) allows ADCG hardware trigger signal to be generated using the internal and external trigger signals output from individual IPs.

29.3.2 Registers

29.3.2.1 List of Registers

PIC2B registers are listed in the following table.

Table 29.42 List of Registers

| Register Name | Symbol | Address |
|--|------------------|--------------------------------|
| A/D converter 0 trigger select control register 0 | PIC2BADCG0TSEL0 | <PIC2B_base> + 00 _H |
| A/D converter 0 trigger select control register 1 | PIC2BADCG0TSEL1 | <PIC2B_base> + 04 _H |
| A/D converter 0 trigger select control register 2 | PIC2BADCG0TSEL2 | <PIC2B_base> + 08 _H |
| A/D converter 0 trigger select control register 3 | PIC2BADCG0TSEL3 | <PIC2B_base> + 0C _H |
| A/D converter 0 trigger select control register 4 | PIC2BADCG0TSEL4 | <PIC2B_base> + 10 _H |
| A/D converter 0 trigger edge select control register | PIC2BADCG0EDGSEL | <PIC2B_base> + 1C _H |
| A/D converter 1 trigger select control register 0 | PIC2BADCG1TSEL0 | <PIC2B_base> + 20 _H |
| A/D converter 1 trigger select control register 1 | PIC2BADCG1TSEL1 | <PIC2B_base> + 24 _H |
| A/D converter 1 trigger select control register 2 | PIC2BADCG1TSEL2 | <PIC2B_base> + 28 _H |
| A/D converter 1 trigger select control register 3 | PIC2BADCG1TSEL3 | <PIC2B_base> + 2C _H |
| A/D converter 1 trigger select control register 4 | PIC2BADCG1TSEL4 | <PIC2B_base> + 30 _H |
| A/D converter 1 trigger edge select control register | PIC2BADCG1EDGSEL | <PIC2B_base> + 3C _H |
| Common to ADCG0, 1 | | |
| A/D converter trigger output control register 400 | PIC2BADTEN400 | <PIC2B_base> + 40 _H |
| A/D converter trigger output control register 401 | PIC2BADTEN401 | <PIC2B_base> + 44 _H |
| A/D converter trigger output control register 402 | PIC2BADTEN402 | <PIC2B_base> + 48 _H |
| A/D converter trigger output control register 403 | PIC2BADTEN403 | <PIC2B_base> + 4C _H |
| A/D converter trigger output control register 404 | PIC2BADTEN404 | <PIC2B_base> + 50 _H |
| A/D converter trigger output control register 410 | PIC2BADTEN410 | <PIC2B_base> + 60 _H |
| A/D converter trigger output control register 411 | PIC2BADTEN411 | <PIC2B_base> + 64 _H |
| A/D converter trigger output control register 412 | PIC2BADTEN412 | <PIC2B_base> + 68 _H |
| A/D converter trigger output control register 413 | PIC2BADTEN413 | <PIC2B_base> + 6C _H |
| A/D converter trigger output control register 414 | PIC2BADTEN414 | <PIC2B_base> + 70 _H |
| A/D converter synchronized start trigger register | ADSYNCTRG | FFF9 1C00 _H |

29.3.2.2 PIC2BADCGnTSELx — A/D Converter n Trigger Select Control Register x

The PIC2BADCGnTSELx register selects triggers for ADCGn scan group x (n = 0, 1; x = 0 to 4).

Access: This register can be read/written in 32-bit units.

Address: Base= FFDD 1000_H
 Base + 00_H (n = 0, x = 0), Base + 04_H (n = 0, x = 1), Base + 08_H (n = 0, x = 2),
 Base + 0C_H (n = 0, x = 3), Base + 10_H (n = 0, x = 4),
 Base + 20_H (n = 1, x = 0), Base + 24_H (n = 1, x = 1), Base + 28_H (n = 1, x = 2),
 Base + 2C_H (n = 1, x = 3), Base + 30_H (n = 1, x = 4)

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| | — | — | ADCGn TSELx 29 | ADCGn TSELx 28 | ADCGn TSELx 27 | ADCGn TSELx 26 | ADCGn TSELx 25 | ADCGn TSELx 24 | — | — | ADCGn TSELx 21 | ADCGn TSELx 20 | ADCGn TSELx 19 | ADCGn TSELx 18 | ADCGn TSELx 17 | ADCGn TSELx 16 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ADCGn TSELx 08 | ADCGn TSELx 07 | ADCGn TSELx 06 | ADCGn TSELx 05 | ADCGn TSELx 04 | ADCGn TSELx 03 | ADCGn TSELx 02 | ADCGn TSELx 01 | ADCGn TSELx 00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.43 PIC2BADCGnTSELx Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 31, 30 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 29 | ADCGn TSELx29 | Selects INTTSG3112 interrupt of TSG31 as the trigger source for ADCGn SGx. 0: INTTSG3112 is not selected. 1: INTTSG3112 is selected. |
| 28 | ADCGn TSELx28 | Selects INTTSG3111 interrupt of TSG31 as the trigger source for ADCGn SGx. 0: INTTSG3111 is not selected. 1: INTTSG3111 is selected. |
| 27 | ADCGn TSELx27 | Selects INTTSG3118 interrupt of TSG31 as the trigger source for ADCGn SGx. 0: INTTSG3118 is not selected. 1: INTTSG3118 is selected. |
| 26 | ADCGn TSELx26 | Selects INTTSG3117 interrupt of TSG31 as the trigger source for ADCGn SGx. 0: INTTSG3117 is not selected. 1: INTTSG3117 is selected. |
| 25 | ADCGn TSELx25 | Selects INTTSG3114 interrupt of TSG31 as the trigger source for ADCGn SGx. 0: INTTSG3114 is not selected. 1: INTTSG3114 is selected. |
| 24 | ADCGn TSELx24 | Selects INTTSG3113 interrupt of TSG31 as the trigger source for ADCGn SGx. 0: INTTSG3113 is not selected. 1: INTTSG3113 is selected. |
| 23, 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | ADCGn TSELx21 | Selects INTTSG3012 interrupt of TSG30 as the trigger source for ADCGn SGx. 0: INTTSG3012 is not selected. 1: INTTSG3012 is selected. |
| 20 | ADCGn TSELx20 | Selects INTTSG3011 interrupt of TSG30 as the trigger source for ADCGn SGx. 0: INTTSG3011 is not selected. 1: INTTSG3011 is selected. |
| 19 | ADCGn TSELx19 | Selects INTTSG3018 interrupt of TSG30 as the trigger source for ADCGn SGx. 0: INTTSG3018 is not selected. 1: INTTSG3018 is selected. |

Table 29.43 PIC2BADCGnTSELx Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 18 | ADCGn TSELx18 | Selects INTTSG30I7 interrupt of TSG30 as the trigger source for ADCGn SGx. 0: INTTSG30I7 is not selected. 1: INTTSG30I7 is selected. |
| 17 | ADCGn TSELx17 | Selects INTTSG30I4 interrupt of TSG30 as the trigger source for ADCGn SGx. 0: INTTSG30I4 is not selected. 1: INTTSG30I4 is selected. |
| 16 | ADCGn TSELx16 | Selects INTTSG30I3 interrupt of TSG30 as the trigger source for ADCGn SGx. 0: INTTSG30I3 is not selected. 1: INTTSG30I3 is selected. |
| 15 to 9 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 8 | ADCGn TSELx08 | Selects ADTRGn as the trigger source for ADCGn SGx. 0: ADTRGn is not selected. 1: ADTRGn is selected. |
| 7 | ADCGn TSELx07 | Selects the TSG3nADTRG1 signal of TSG31 as the trigger source for ADCGn SGx. 0: TSG3nADTRG1 is not selected. 1: TSG3nADTRG1 is selected. |
| 6 | ADCGn TSELx06 | Selects the TSG3nADTRG0 signal of TSG31 as the trigger source for ADCGn SGx. 0: TSG3nADTRG0 is not selected. 1: TSG3nADTRG0 is selected. |
| 5 | ADCGn TSELx05 | Selects the TSG3nADTRG1 signal of TSG30 as the trigger source for ADCGn SGx. 0: TSG3nADTRG1 is not selected. 1: TSG3nADTRG1 is selected. |
| 4 | ADCGn TSELx04 | Selects the TSG3nADTRG0 signal of TSG30 as the trigger source for ADCGn SGx. 0: TSG3nADTRG0 is not selected. 1: TSG3nADTRG0 is selected. |
| 3 | ADCGn TSELx03 | Selects the INTENCA1I1 interrupt signal of ENCA1 as the trigger source for ADCGn SGx. 0: INTENCA1I1 is not selected. 1: INTENCA1I1 is selected. |
| 2 | ADCGn TSELx02 | Selects the INTENCA0I1 interrupt signal of ENCA0 as the trigger source for ADCGn SGx. 0: INTENCA0I1 is not selected. 1: INTENCA0I1 is selected. |
| 1 | ADCGn TSELx01 | Selects the TAUD1 interrupt signal selected by the PIC2BADTEN41x register as the trigger source for ADCGn SGx. 0: TAUD1 interrupt is not selected. 1: TAUD1 interrupt is selected. |
| 0 | ADCGn TSELx00 | Selects the TAUD0 interrupt signal selected by the PIC2BADTEN40x register as the trigger source for ADCGn SGx. 0: TAUD0 interrupt is not selected. 1: TAUD0 interrupt is selected. |

29.3.2.3 PIC2BADCGnEDGSEL — A/D Converter Trigger Edge Control Register

The PIC2BADCGnEDGSEL register selects an effective edge for the one-shot pulse generation circuit which generates an AD CG trigger.

Access: This register can be read/written in 16-bit units.

Address: Base = FFDD 1000_H
Base + 1C_H (n = 0), Base + 3C_H (n = 1)

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|-----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | PIC2BADCGnEDGSEL[9:0] | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.44 PIC2BADCGnEDGSEL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------|---|
| 15 to 10 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 9, 8 | PIC2BADCGnEDGSEL[9:8] | Selects an effective edge of AD CGn scan group 4. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.) |
| 7, 6 | PIC2BADCGnEDGSEL[7:6] | Selects an effective edge of AD CGn scan group 3. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.) |
| 5, 4 | PIC2BADCGnEDGSEL[5:4] | Selects an effective edge of AD CGn scan group 2. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.) |
| 3, 2 | PIC2BADCGnEDGSEL[3:2] | Selects an effective edge of AD CGn scan group 1. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.) |
| 1, 0 | PIC2BADCGnEDGSEL[1:0] | Selects an effective edge of AD CGn scan group 0. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.) |

29.3.2.4 PIC2BADTEN4nx — A/D Converter Trigger Output Select Control Register

The PIC2BADTEN4nx register enables selecting a trigger source from TAUDn channel m as the ADCG trigger. (n = 0, 1; x = 0 to 4). This register is common to ADCG0 and ADCG1.

Access: This register can be read/written in 16-bit units.

Address: Base = FFDD 1000_H
 Base + 40_H (n = 0, x = 0), Base + 44_H (n = 0, x = 1), Base + 48_H (n = 0, x = 2),
 Base + 4C_H (n = 0, x = 3), Base + 50_H (n = 0, x = 4),
 Base + 60_H (n = 1, x = 0), Base + 64_H (n = 1, x = 1), Base + 68_H (n = 1, x = 2),
 Base + 6C_H (n = 1, x = 3), Base + 70_H (n = 1, x = 4)

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | PIC2BADTEN4nx[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 29.45 PIC2BADTEN4nx Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------------|---|
| 15 to 0 | PIC2BADTEN4nxm (m = 0 to 15) | Set a trigger source from TAUDn channel m. 0: Trigger source of TAUDn channel m cannot be selected as the ADCG trigger. 1: Trigger source of TAUDn channel m can be selected as the ADCG trigger. |

29.3.2.5 ADSYNCTRГ — A/D Converter Synchronized Start Trigger Register

The ADSYNCTRГ register controls synchronized start trigger for ADCG0 and ADCG1.

Access: This register can be read/written in 8-bit units.

Address: FFF9 1C00_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|------------|------------|------------|------------|------------|
| | — | — | — | ADSYNCTRГ4 | ADSYNCTRГ3 | ADSYNCTRГ2 | ADSYNCTRГ1 | ADSYNCTRГ0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 29.46 ADSYNCTRГ Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | ADSYNCTRГ4 | Synchronously converts the SG of ADCG1 with the conversion start trigger selected by the individual SG of ADCG0. |
| 3 | ADSYNCTRГ3 | 0: Synchronized start is disabled. |
| 2 | ADSYNCTRГ2 | 1: Synchronized start is enabled. |
| 1 | ADSYNCTRГ1 | The function only applies to normal A/D conversion of ADCG0 and ADCG1. |
| 0 | ADSYNCTRГ0 | |

29.3.3 Function

29.3.3.1 ADCG Trigger Select Function

(1) Overview

The function allows generation of ADCG hardware trigger signal for individual scan groups by the signals from each IP. The IPs can be selected from TAUD0, TAUD1, ENCA0, ENCA1, TSG30, and TSG31.

(2) Configuration

The ADCG trigger select function is realized by using individual IPs and PIC2B in combination. The following figure shows the block diagram of ADCG trigger select function.

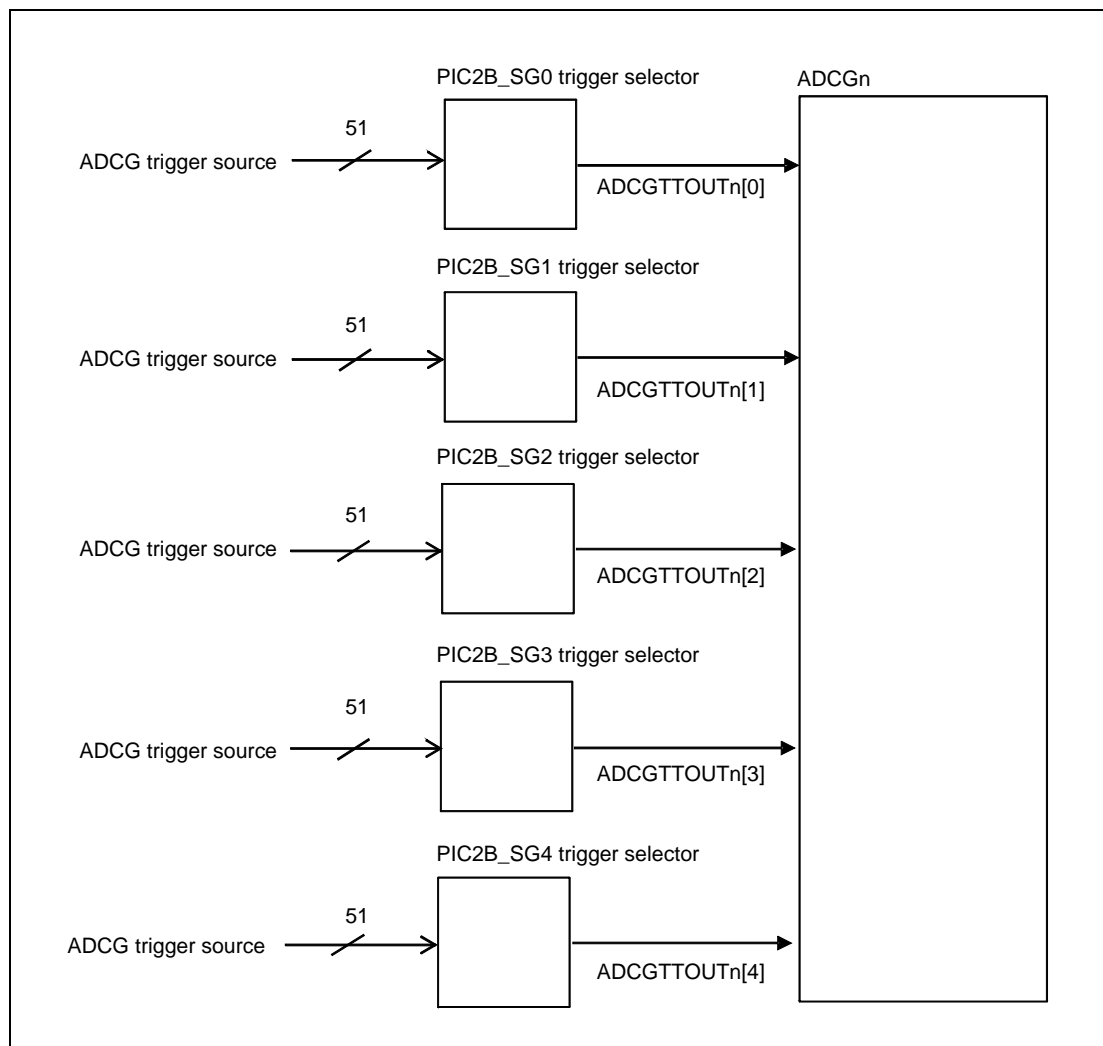


Figure 29.47 Block Diagram of ADCG Trigger Select Function

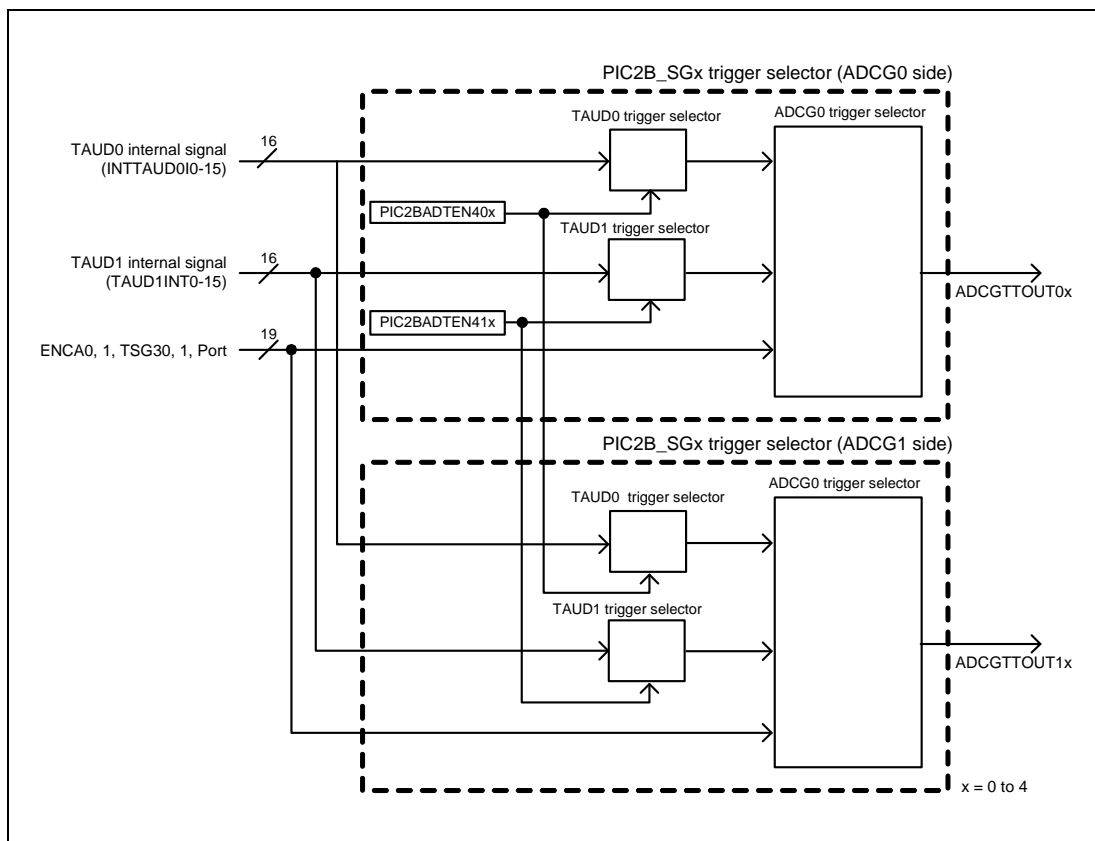


Figure 29.48 Block Diagram of PIC2B_SGx Trigger Selection

(3) Registers

For the settings of the registers used in this function, see **Figure 29.49, Block Diagram of PIC2B**, and from **Section 29.3.2.2** to **Section 29.3.2.4**.

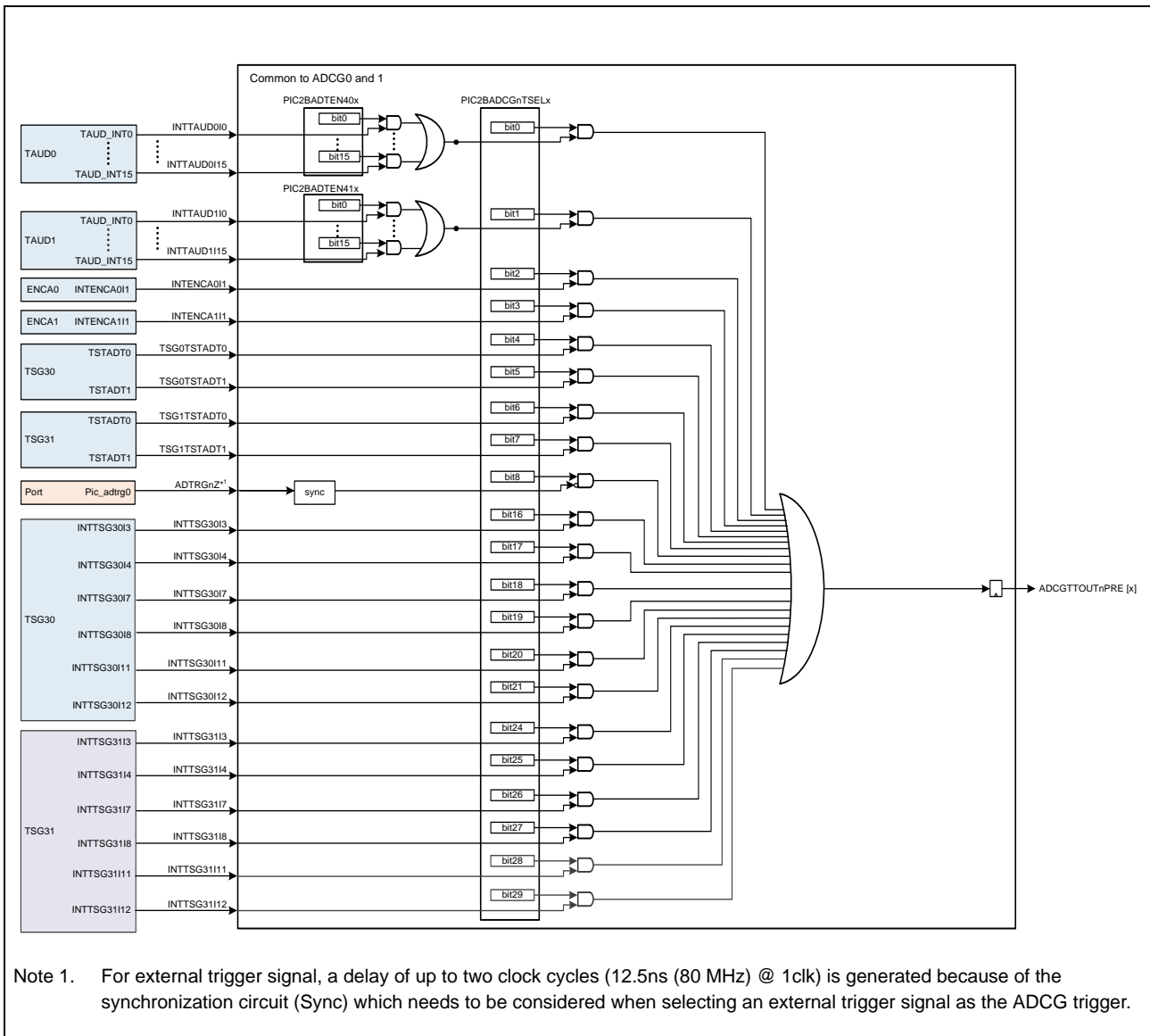


Figure 29.49 Block Diagram of PIC2B

(4) Function

A/D trigger signal can be selected for individual ADCGn scan groups and output can be selected from rising edge, falling edge, and both edges. The TAUD trigger for the scan group with the same number is shared between ADCG0 and ADCG1.

(5) Flow Chart

Set this function before starting A/D converter.

Section 30 A/D Converter (ADCG)

The A/D Converter (ADCG) is advanced A/D converter of successive approximation method. ADCG has two units implemented.

30.1 Features of RH850/P1M-E ADCG

30.1.1 Number of Units

This microcontroller has the following number of ADCG units.

Table 30.1 Number of Units

| Products | RH850/P1M-E |
|-----------------|------------------|
| Number of units | 2 |
| Name | ADCGn (n = 0, 1) |

Table 30.2 Index

| Index | Meaning |
|-------|--|
| n | Throughout this section, the individual ADCG units are identified by the index "n" (n = 0, 1). |
| m | Throughout this section, the number of each ADCG physical channel is indicated by the index "m" (see Table 30.6 for physical channel number m). |
| j | Throughout this section, the number of data registers and virtual channels are identified by the index "j" (j = 0 to 23), for example, ADCGnDRj for the data register j. |
| k | Throughout this section, the T&H channel number is indicated by the letter "k" (k = 0 to 5 for ADCG0; k = 0 to 3 for ADCG1). |
| x | Throughout this section, the scan group is indicated by the letter "x" (x = 0 to 4). |
| y | Throughout this section, the number of A/D timers are indicated by the letter "y" (y = 3, 4). |
| z | Throughout this section, A/D conversion monitor virtual channel pointer is indicated by the index "z" (z = 0 to 4). |

30.1.2 Register Base Address

ADCG base addresses are listed in the following table.

ADCG register addresses are given as offsets from the individual base address.

Table 30.3 Register Base Address

| Base Address Name | Base Address |
|-------------------|------------------------|
| <ADCG0_base> | FFF9 1000 _H |
| <ADCG1_base> | FFF9 2000 _H |

30.1.3 Clock Supply

Clock supply by and to ADCG is listed in the following table.

Table 30.4 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|------------------------------------|
| ADCGn | PCLK | Low-speed peripheral clock CLK_LSB |
| | ADC clock | ADCG clock CLK_ADC |

30.1.4 Interrupt Requests

ADCG interrupt requests are listed in the following table.

Table 30.5 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Code | DMA/DTS Trigger | ECM |
|---------------------|----------------------------------|----------------|-----------------|-----|
| ADCG0 | | | | |
| INTADCG0I0 | ADCG0 scan group 0 end interrupt | √ | √ | — |
| INTADCG0I1 | ADCG0 scan group 1 end interrupt | √ | √ | — |
| INTADCG0I2 | ADCG0 scan group 2 end interrupt | √ | √ | — |
| INTADCG0I3 | ADCG0 scan group 3 end interrupt | √ | √ | — |
| INTADCG0I4 | ADCG0 scan group 4 end interrupt | √ | √ | — |
| INTADCG0ERR | ADCG0 A/D error interrupt | √ | — | — |
| INTADCG0MPX | ADCG0 MPX request interrupt | √ | √ | — |
| INTADCG0PE | ADCG0 parity error interrupt | — | — | √ |
| ADCG1 | | | | |
| INTADCG1I0 | ADCG1 scan group 0 end interrupt | √ | √ | — |
| INTADCG1I1 | ADCG1 scan group 1 end interrupt | √ | √ | — |
| INTADCG1I2 | ADCG1 scan group 2 end interrupt | √ | √ | — |
| INTADCG1I3 | ADCG1 scan group 3 end interrupt | √ | √ | — |
| INTADCG1I4 | ADCG1 scan group 4 end interrupt | √ | √ | — |
| INTADCG1ERR | ADCG1 A/D error interrupt | √ | — | — |
| INTADCG1MPX | ADCG1 MPX request interrupt | √ | √ | — |
| INTADCG1PE | ADCG1 parity error interrupt | — | — | √ |

30.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

30.1.6 External Input/Output Signals

External input/output signals of ADCG are listed in the following table.

Table 30.6 External Input/Output Signals

| Unit Signal Name | I/O | Description | Alternative Port Signal Name |
|------------------|-----|--|------------------------------|
| AVcc0 | — | Power supply pin for the input analog part | A0VCC |
| AVss0 | — | Ground pin for the input analog part | A0VSS |
| AVcc1 | — | Power supply pin for the input analog part | A1VCC |
| AVss1 | — | Ground pin for the input analog part | A1VSS |
| AVREFH0 | I | Reference voltage pin for the input analog part | A0VREFH |
| AVREFH1 | I | Reference voltage pin for the input analog part | A1VREFH |
| AN000 | I | 12-bit resolution analog input pin (for T&H) | ADCG0I0 |
| AN001 | I | 12-bit resolution analog input pin (for T&H) | ADCG0I1 |
| AN002 | I | 12-bit resolution analog input pin (for T&H) | ADCG0I2 |
| AN003 | I | 12-bit resolution analog input pin (for T&H) | ADCG0I3 |
| AN004 | I | 12-bit resolution analog input pin (for T&H) | ADCG0I4 |
| AN005 | I | 12-bit resolution analog input pin (for T&H) | ADCG0I5 |
| AN006 | I | 12-bit resolution analog input pin (for MPX) | ADCG0I6 |
| AN007 | I | 12-bit resolution analog input pin | ADCG0I7 |
| AN008 | I | 12-bit resolution analog input pin | ADCG0I8 |
| AN009 | I | 12-bit resolution analog input pin | ADCG0I9*1 |
| AN010 | I | 12-bit resolution analog input pin | ADCG0I10*1 |
| AN011 | I | 12-bit resolution analog input pin | ADCG0I11*1 |
| AN100 | I | 12-bit resolution analog input pin (for T&H and MPX) | ADCG1I0 |
| AN101 | I | 12-bit resolution analog input pin (for T&H) | ADCG1I1 |
| AN102 | I | 12-bit resolution analog input pin (for T&H) | ADCG1I2 |
| AN103 | I | 12-bit resolution analog input pin (for T&H) | ADCG1I3 |
| AN104 | I | 12-bit resolution analog input pin | ADCG1I4 |
| AN105 | I | 12-bit resolution analog input pin | ADCG1I5 |
| AN106 | I | 12-bit resolution analog input pin | ADCG1I6 |
| AN107 | I | 12-bit resolution analog input pin | ADCG1I7 |
| AN108 | I | 12-bit resolution analog input pin | ADCG1I8 |
| AN109 | I | 12-bit resolution analog input pin | ADCG1I9 |
| AN110 | I | 12-bit resolution analog input pin | ADCG1I10*1 |
| AN111 | I | 12-bit resolution analog input pin | ADCG1I11*1 |
| AN112 | — | Input channel for internal temperature sensor | — |
| ADTRG0 | I | External input trigger pin | ADCGTRG0 |
| ADTRG1 | I | External input trigger pin | ADCGTRG1 |
| ADCG0CNV0 | O | A/D converter status monitor pin | ADCG0CNV0 |
| ADCG0CNV1 | O | A/D converter status monitor pin | ADCG0CNV1 |
| ADCG0CNV2 | O | A/D converter status monitor pin | ADCG0CNV2 |
| ADCG0CNV3 | O | A/D converter status monitor pin | ADCG0CNV3 |
| ADCG0CNV4 | O | A/D converter status monitor pin | ADCG0CNV4 |
| ADCG1CNV0 | O | A/D converter status monitor pin | ADCG1CNV0 |
| ADCG1CNV1 | O | A/D converter status monitor pin | ADCG1CNV1 |
| ADCG1CNV2 | O | A/D converter status monitor pin | ADCG1CNV2 |
| ADCG1CNV3 | O | A/D converter status monitor pin | ADCG1CNV3 |
| ADCG1CNV4 | O | A/D converter status monitor pin | ADCG1CNV4 |

Note 1. This channel is not supported in this product (100 pin).

30.1.7 Analog Channels and Track and Hold Function

A/D conversion is available for 12 ADCG0 channels and 13 ADCG1 channels (including the temperature sensor dedicated channel), 25 channels in total. For corresponding channel numbers for each product, see **Table 30.6**.

ADCG0 and ADCG1 have an internal track and hold circuit that enables A/D conversion with simultaneous sampling of the voltages on different channels.

ADCG0 has a track and hold circuit for six channels that enables simultaneous track and hold for up to six channels.

ADCG1 has a track and hold circuit for four channels that enables simultaneous track and hold for up to four channels.

30.1.8 Virtual Channel

Each ADCG has 24 virtual channels. Analog channels for which A/D conversion is to be made and other accompanying information are configured for each virtual channel. By sequentially performing the processing for the virtual channels indicated by the start virtual channel pointer and the end virtual channel pointer in each scan group, scans (which can perform A/D conversion for any analog channels in any order) can be executed.

30.2 Overview

30.2.1 Functional Overview

ADCG has the following features.

Advanced A/D converter

Resolution: 12 bits

A/D conversion method: Successive approximation method

A/D conversion time: Selectable as 1.0 μ s and 11.3 μ s.

(The A/D conversion time is the sum of the sampling time of the sample-and-hold circuits and the time for conversion by successive approximation.)

- Supports five scan groups
Each ADCG has five scan groups. Scan settings can be made independently for each scan group.
- Two scan modes
Each ADCG has two scan modes.
Multicycle scan mode: Specified number of scans are executed.
Continuous scan mode: Scans are repeatedly executed without limit.
- Interval function
The ADCG can start scan groups in any cycle by using the A/D timer equipped in the scan groups 3 and 4. This enables scans with intervals inserted.
- A/D-converted value adding function
The ADCG performs A/D conversion sequentially twice or four times for a channel, and stores the addition result in the data register. The addition count setting is common to all virtual channels. The effect of the moving average filter can be gained by using this result. However, this function does not always ensure that A/D conversion accuracy is improved.
- Extended physical channels
Each ADCG can extend physical channels by using an external analog multiplexer. (Available channels are AN006 and AN100.)
- Data registers
Data registers corresponding to virtual channels are provided.
- Start trigger for each scan group
Hardware triggers and software triggers can start processing of each scan group. Only scan groups 3 and 4 can start processing by an A/D timer trigger.
- Asynchronous/synchronous suspend and resume function
A processing for a scan group can interrupt an ongoing processing for another scan group. The priority is as follows:

| | |
|-----------------------------|------------------|
| Low | High |
| SG0 < SG1 < SG2 < SG3 < SG4 | (SG: Scan group) |

If a request for a higher-priority SG is made while a lower-priority SG is being processed, the lower-priority SG is suspended after the ongoing virtual channel processing is stopped (synchronous suspend) or after the ongoing virtual channel processing is immediately stopped (asynchronous suspend), and then the processing for the higher-priority SG is performed. After the processing for the higher-priority SG is completed, the suspended virtual channel processing of the lower-priority SG resumes (synchronous suspend is available only when T&H function is not used.)

The following configuration is also available: when a higher-priority SG interrupts an SG0 processing, asynchronous suspend occurs, but when a higher-priority SG interrupts a lower-priority SG other than SG0, synchronous suspend occurs (only when T&H function is not used).

- Supports scan end interrupt and DMA transfer
Each scan group can generate an interrupt request to the INTC and activate the DMAC each time a processing for the virtual channel indicated by the end virtual channel pointer ends or a virtual channel ends.
- An analog conversion voltage can be specified
The A0VREFH pin and the A1VREFH pin can be used to set the voltage range for analog conversion.
- A variety of safety functions
The ADCG is equipped with various safety functions, including A/D conversion circuit selfdiagnostic function, pin-level self-diagnostic function, wiring-break detection, ID error check, upper-limit/lower-limit check for data registers, parity check for data registers, overwrite check for data registers, and read and clear function for data registers.

30.2.2 Block Diagram

Figure 30.1 and Figure 30.2 show the ADCG block diagram.

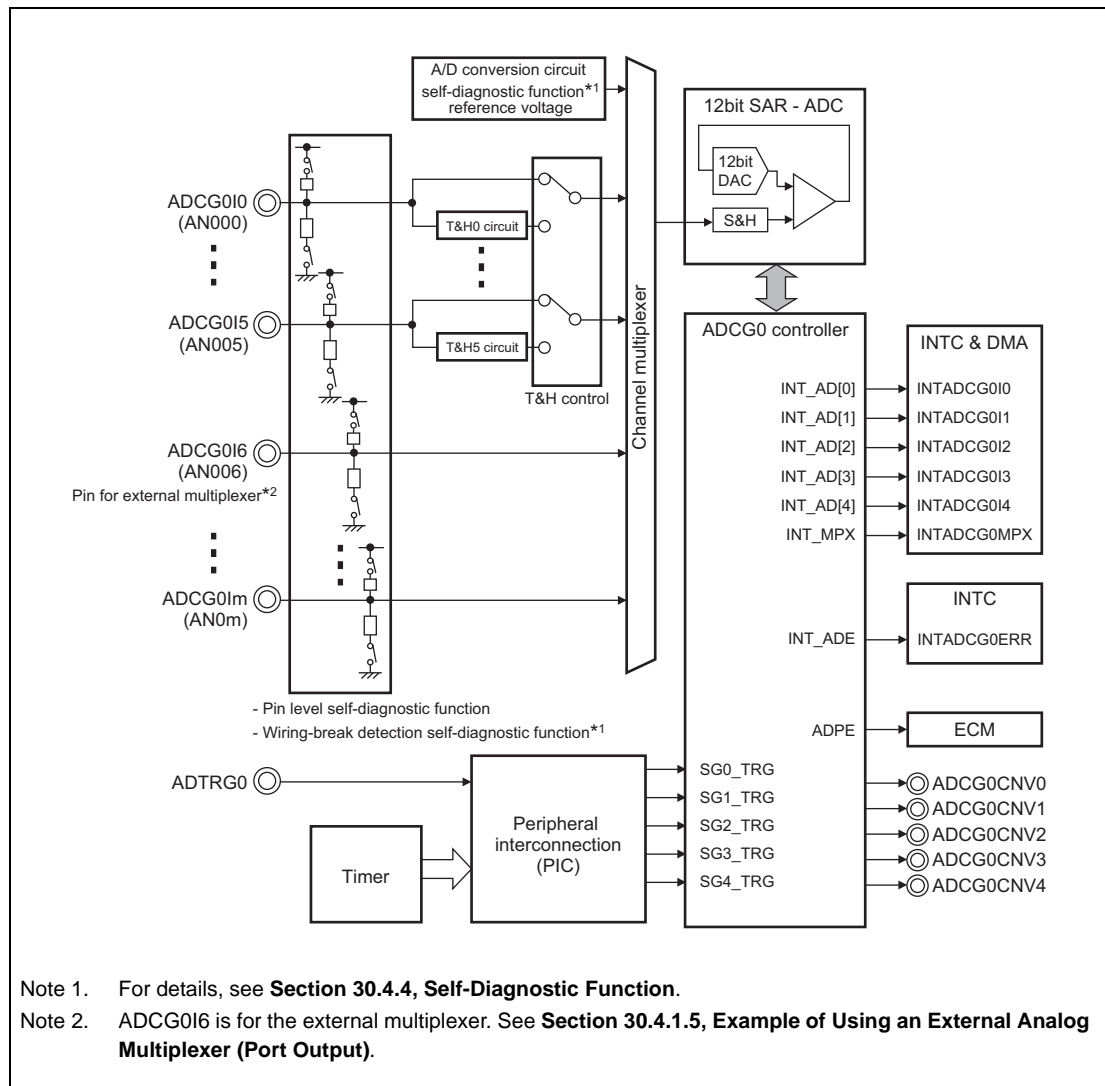


Figure 30.1 Block Diagram of ADCG0

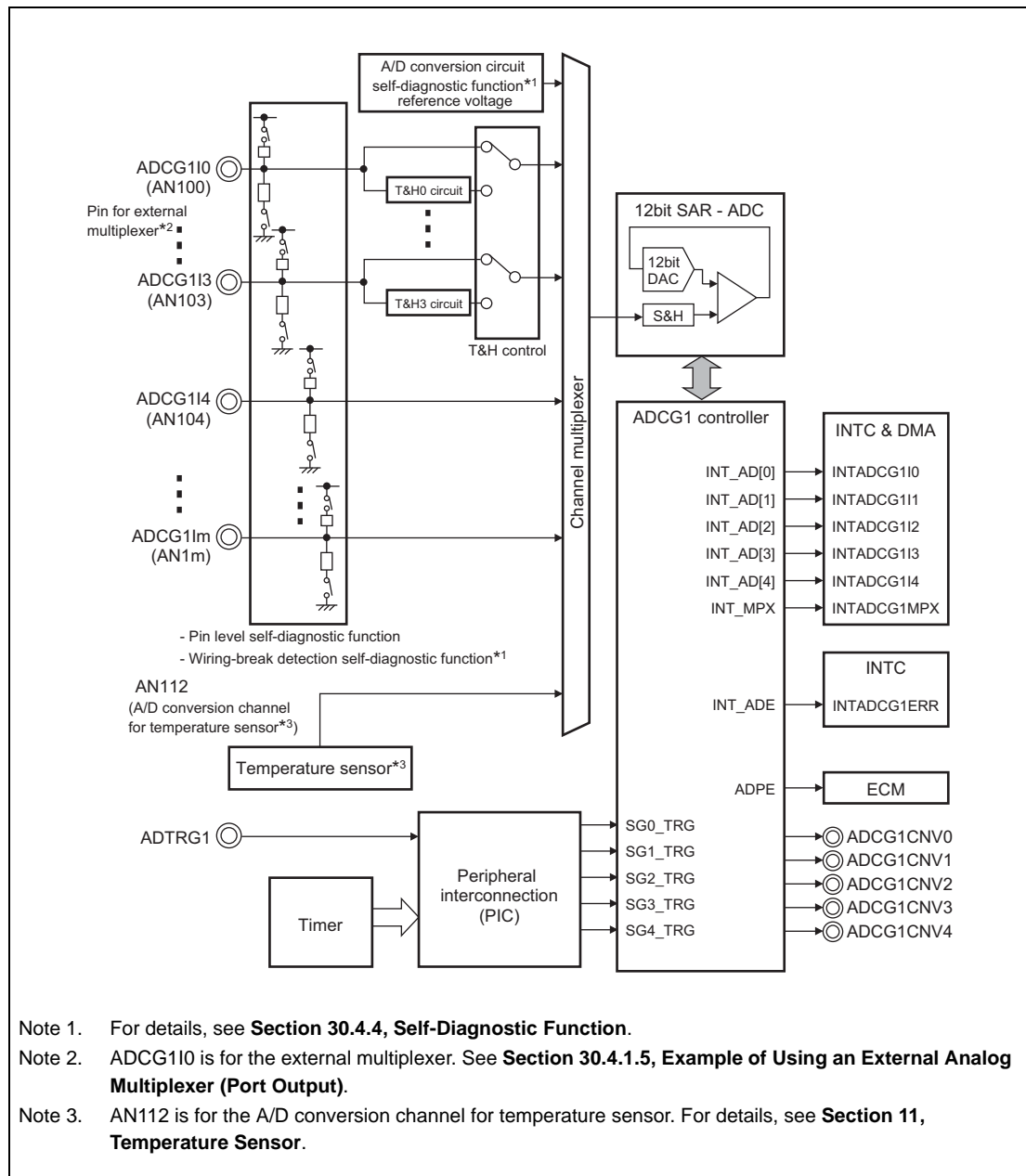


Figure 30.2 Block Diagram of ADCG1

30.2.3 Scan Group (SG)

A scan group is a group of multiple virtual channels.

ADCG has 5 scan groups and the order of priority for A/D conversion is as follows: SG4 > SG3 > SG2 > SG1 > SG0.

SGx can group consecutive virtual channels. Grouping is set up by using a start pointer (ADCGnSGVCSPx register) and an end pointer (ADCGnSGVCEPx register). Disable the A/D conversion trigger input for any unused scan groups.

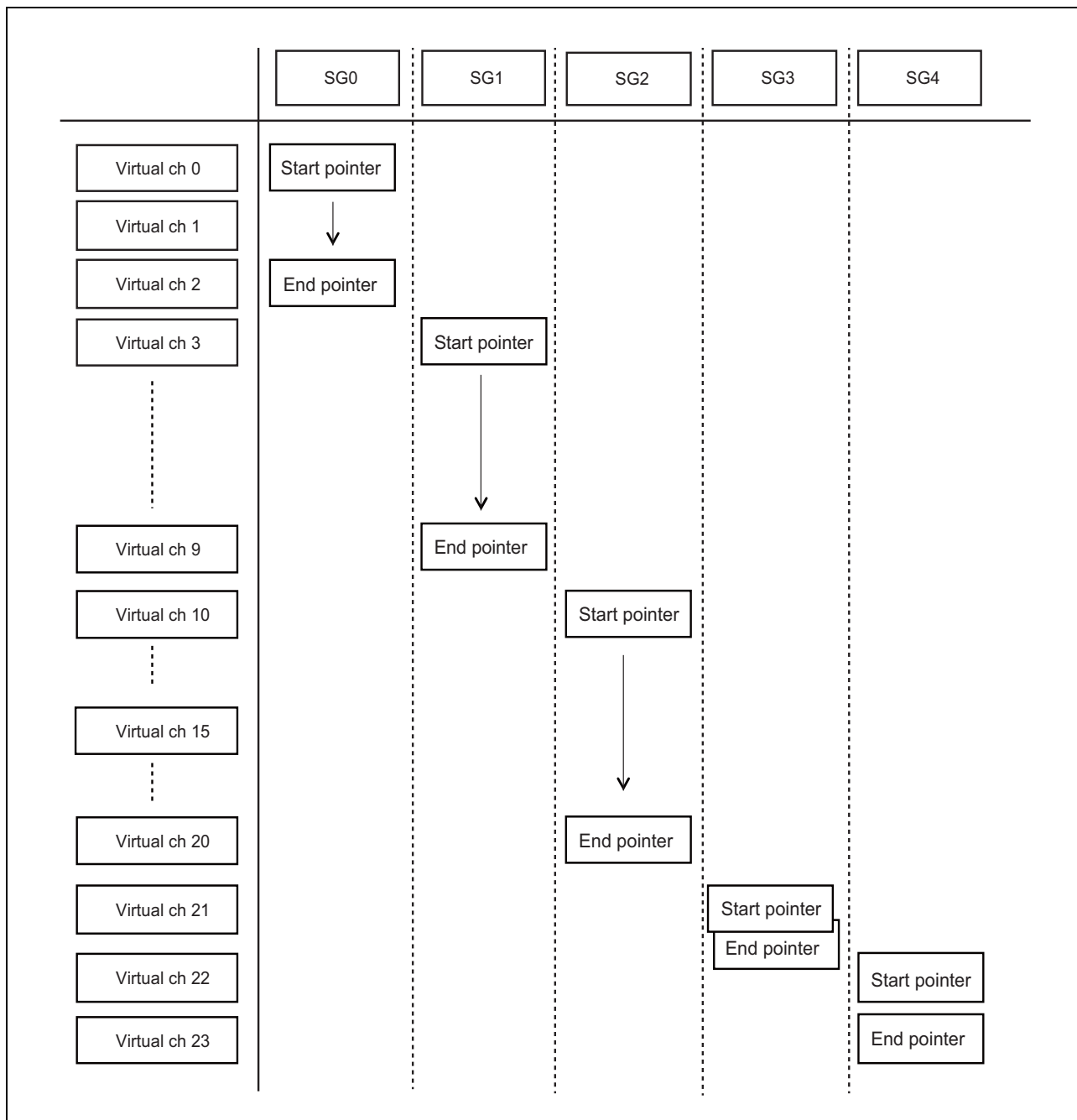


Figure 30.3 Example of SG Assignment

30.3 Registers

30.3.1 List of Registers

ADCG registers are listed in the following table.

Table 30.7 List of Registers (1/2)

| Register Name | Symbol | Address |
|--|-----------------|--|
| ADC common registers | | |
| A/D synchronization start control register | ADCG0ADSYNSTCR | <ADCG0_base> + 300 _H |
| A/D timer synchronization start control register | ADCG0ADTSYNSTCR | <ADCG0_base> + 304 _H |
| ADC specific registers (virtual channel) | | |
| Virtual channel register j | ADCGnVCRj | <ADCGn_base> + j × 4 _H |
| Data register j (j = only even numbers) | ADCGnDRj | <ADCGn_base> + 100 _H + j × 2 _H |
| Data supplementary information register j | ADCGnDIRj | <ADCGn_base> + 200 _H + j × 4 _H |
| ADC specific registers (control) | | |
| A/D conversion time control register | ADCGnSMPCR | <ADCGn_base> + 340 _H |
| A/D halt register | ADCGnADHALTR | <ADCGn_base> + 380 _H |
| A/D control register 1 | ADCGnADCR1 | <ADCGn_base> + 384 _H |
| MPX current control register | ADCGnMPXCURCR | <ADCGn_base> + 388 _H |
| MPX current register | ADCGnMPXCURR | <ADCGn_base> + 38C _H |
| MPX optional wait register | ADCGnMPXOWR | <ADCGn_base> + 390 _H |
| A/D control register 2 | ADCGnADCR2 | <ADCGn_base> + 398 _H |
| A/D conversion monitor virtual channel pointer 0 | ADCGnADENDP0 | <ADCGn_base> + 3A0 _H |
| A/D conversion monitor virtual channel pointer 1 | ADCGnADENDP1 | <ADCGn_base> + 3A4 _H |
| A/D conversion monitor virtual channel pointer 2 | ADCGnADENDP2 | <ADCGn_base> + 3A8 _H |
| A/D conversion monitor virtual channel pointer 3 | ADCGnADENDP3 | <ADCGn_base> + 3AC _H |
| A/D conversion monitor virtual channel pointer 4 | ADCGnADENDP4 | <ADCGn_base> + 3B0 _H |
| T&H sampling start control register | ADCGnTHSMPSTCR | <ADCGn_base> + 400 _H |
| T&H stop control register | ADCGnTHSTPCR | <ADCGn_base> + 404 _H |
| T&H control register | ADCGnTHCR | <ADCGn_base> + 408 _H |
| T&H group A hold start control register | ADCGnTHAHLSTCR | <ADCGn_base> + 410 _H |
| T&H group B hold start control register | ADCGnTHBHLSTCR | <ADCGn_base> + 414 _H |
| T&H group A control register | ADCGnTHACR | <ADCGn_base> + 420 _H |
| T&H group B control register | ADCGnTHBCR | <ADCGn_base> + 424 _H |
| T&H enable register | ADCGnTHER | <ADCGn_base> + 430 _H |
| T&H group select register | ADCGnTHGSR | <ADCGn_base> + 434 _H |
| ADC specific registers (safety-related) | | |
| Safety control register | ADCGnSFTCR | <ADCGn_base> + 3C0 _H |
| Pin level self-diagnostic control register | ADCGnTDCR | <ADCGn_base> + 3C4 _H |
| Wiring-break detection control register | ADCGnODCR | <ADCGn_base> + 3C8 _H |
| Wiring-break detection pin setting register 0 | ADOPDIG0 | FFF9 1800 _H |
| Wiring-break detection pin setting register 1 | ADOPDIG1 | FFF9 2800 _H |
| Upper-limit/lower-limit table register 0 | ADCGnULLMTBR0 | <ADCGn_base> + 3CC _H |
| Upper-limit/lower-limit table register 1 | ADCGnULLMTBR1 | <ADCGn_base> + 3D0 _H |
| Upper-limit/lower-limit table register 2 | ADCGnULLMTBR2 | <ADCGn_base> + 3D4 _H |
| Error clear register | ADCGnECR | <ADCGn_base> + 3D8 _H |
| Upper-limit/lower-limit error register | ADCGnULER | <ADCGn_base> + 3DC _H |
| Overwrite error register | ADCGnOWER | <ADCGn_base> + 3E0 _H |
| Parity error register | ADCGnPER | <ADCGn_base> + 3E4 _H |

Table 30.7 List of Registers (2/2)

| Register Name | Symbol | Address |
|--|----------------|---------------------------------|
| ID error register | ADCGnIDER | <ADCGn_base> + 3E8 _H |
| Scan group specific registers | | |
| Scan group x start control register 0 | ADCGnSGSTCR0 | <ADCGn_base> + 480 _H |
| Scan group x start control register 1 | ADCGnSGSTCR1 | <ADCGn_base> + 500 _H |
| Scan group x start control register 2 | ADCGnSGSTCR2 | <ADCGn_base> + 580 _H |
| Scan group x start control register 3 | ADCGnSGSTCR3 | <ADCGn_base> + 600 _H |
| Scan group x start control register 4 | ADCGnSGSTCR4 | <ADCGn_base> + 680 _H |
| A/D timer y start control register 3 | ADCGnADTSTCR3 | <ADCGn_base> + 608 _H |
| A/D timer y start control register 4 | ADCGnADTSTCR4 | <ADCGn_base> + 688 _H |
| A/D timer y end control register 3 | ADCGnADTENDCR3 | <ADCGn_base> + 60C _H |
| A/D timer y end control register 4 | ADCGnADTENDCR4 | <ADCGn_base> + 68C _H |
| Scan group x control register 0 | ADCGnSGCR0 | <ADCGn_base> + 490 _H |
| Scan group x control register 1 | ADCGnSGCR1 | <ADCGn_base> + 510 _H |
| Scan group x control register 2 | ADCGnSGCR2 | <ADCGn_base> + 590 _H |
| Scan group x control register 3 | ADCGnSGCR3 | <ADCGn_base> + 610 _H |
| Scan group x control register 4 | ADCGnSGCR4 | <ADCGn_base> + 690 _H |
| Scan group x start virtual channel pointer 0 | ADCGnSGVCSP0 | <ADCGn_base> + 494 _H |
| Scan group x start virtual channel pointer 1 | ADCGnSGVCSP1 | <ADCGn_base> + 514 _H |
| Scan group x start virtual channel pointer 2 | ADCGnSGVCSP2 | <ADCGn_base> + 594 _H |
| Scan group x start virtual channel pointer 3 | ADCGnSGVCSP3 | <ADCGn_base> + 614 _H |
| Scan group x start virtual channel pointer 4 | ADCGnSGVCSP4 | <ADCGn_base> + 694 _H |
| Scan group x end virtual channel pointer 0 | ADCGnSGVCEP0 | <ADCGn_base> + 498 _H |
| Scan group x end virtual channel pointer 1 | ADCGnSGVCEP1 | <ADCGn_base> + 518 _H |
| Scan group x end virtual channel pointer 2 | ADCGnSGVCEP2 | <ADCGn_base> + 598 _H |
| Scan group x end virtual channel pointer 3 | ADCGnSGVCEP3 | <ADCGn_base> + 618 _H |
| Scan group x end virtual channel pointer 4 | ADCGnSGVCEP4 | <ADCGn_base> + 698 _H |
| Scan group x multicycle register 0 | ADCGnSGMCYCR0 | <ADCGn_base> + 49C _H |
| Scan group x multicycle register 1 | ADCGnSGMCYCR1 | <ADCGn_base> + 51C _H |
| Scan group x multicycle register 2 | ADCGnSGMCYCR2 | <ADCGn_base> + 59C _H |
| Scan group x multicycle register 3 | ADCGnSGMCYCR3 | <ADCGn_base> + 61C _H |
| Scan group x multicycle register 4 | ADCGnSGMCYCR4 | <ADCGn_base> + 69C _H |
| Scan group x status register 0 | ADCGnSGSR0 | <ADCGn_base> + 4A4 _H |
| Scan group x status register 1 | ADCGnSGSR1 | <ADCGn_base> + 524 _H |
| Scan group x status register 2 | ADCGnSGSR2 | <ADCGn_base> + 5A4 _H |
| Scan group x status register 3 | ADCGnSGSR3 | <ADCGn_base> + 624 _H |
| Scan group x status register 4 | ADCGnSGSR4 | <ADCGn_base> + 6A4 _H |
| A/D timer initial phase register 3 | ADCGnADTIPR3 | <ADCGn_base> + 628 _H |
| A/D timer initial phase register 4 | ADCGnADTIPR4 | <ADCGn_base> + 6A8 _H |
| A/D timer period register 3 | ADCGnADTPRR3 | <ADCGn_base> + 62C _H |
| A/D timer period register 4 | ADCGnADTPRR4 | <ADCGn_base> + 6AC _H |
| Scan group x upper-limit/lower-limit table select register 0 | ADCGnULLMSR0 | <ADCGn_base> + 4B0 _H |
| Scan group x upper-limit/lower-limit table select register 1 | ADCGnULLMSR1 | <ADCGn_base> + 530 _H |
| Scan group x upper-limit/lower-limit table select register 2 | ADCGnULLMSR2 | <ADCGn_base> + 5B0 _H |
| Scan group x upper-limit/lower-limit table select register 3 | ADCGnULLMSR3 | <ADCGn_base> + 630 _H |
| Scan group x upper-limit/lower-limit table select register 4 | ADCGnULLMSR4 | <ADCGn_base> + 6B0 _H |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.2 ADCG0ADSYNSTCR — A/D Synchronization Start Control Register

ADCG0ADSYNSTCR is an 8-bit write-only register that controls simultaneous start of A/D conversion for each scan group of ADCG0 and ADCG1. The register bits are always read as 0.

Access: This register can be written in 8-bit units.

Address: <ADCG0_base> + 300_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ADSTART |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.8 ADCG0ADSYNSTCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ADSTART | Starts simultaneous A/D conversion for scan groups of ADCG0 and ADCG1. 0: No function. (Writing 0 is ignored.) 1: A/D conversion is started simultaneously. Enable the SG synchronization start enable bit (ADCGnSGCRx.ADSTARTE) for the scan group which is to have A/D conversion simultaneously started. |

30.3.3 ADCG0ADTSYNSTCR — A/D Timer Synchronization Start Control Register

ADCG0ADTSYNSTCR is an 8-bit write-only register that controls simultaneous start of count operation for each A/D timer of ADCG0 and ADCG1. The register bits are always read as 0.

Access: This register can be written in 8-bit units.

Address: <ADCG0_base> + 304_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ADTSTART |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.9 ADCG0ADTSYNSTCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ADTSTART | Starts count operation of the A/D timer of ADCG0 and ADCG1 simultaneously. 0: No function. (Writing 0 is ignored.) 1: A/D timer count is started. Enable the A/D timer synchronization start enable bit (the ADCGnSGCRx.ADTSTARTE) for the A/D timer which is to have count operation simultaneously started. |

30.3.4 ADCGnVCRj — Virtual Channel Register j

ADCGnVCRj is a 32-bit readable/writable register configured for each virtual channel.

Access: This register can be read/written in 32-bit units.

Address: <ADCGn_base> + j × 4_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PUE | PDE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------------|-----|-----|----|----|----|---|---|------|---|------------|-----|-----|-----|-----|-----|
| | CNVCLS[2:0] | | | — | — | — | — | — | ADIE | — | GCTRL[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.10 ADCGnVCRj Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 17 | PUE | <p>Pull-up Resistor Control for Physical I/O Channels</p> <p>0: Pull-up resistor for the target physical I/O channel is off.</p> <p>1: Pull-up resistor for the target physical I/O channel is on.</p> <p>When PUE = 1, this bit turns on the pull-up resistor of the physical I/O channel specified for this virtual channel when the wiring-break detection is in use.</p> <p>CAUTION</p> <ol style="list-style-type: none"> Do not set the PDE and PUE bits at the same time. Do not set the held value A/D conversion while PUE = 1. Do not use the pin-level self-diagnosis while PUE = 1. |
| 16 | PDE | <p>Pull-down Resistor Control for Physical I/O Channels</p> <p>0: Pull-down resistor for the target physical I/O channel is off.</p> <p>1: Pull-down resistor for the target physical I/O channel is on.</p> <p>When PDE = 1, this bit turns on the pull-down resistor of the physical I/O channel specified for this virtual channel when the wiring-break detection is in use.</p> <p>CAUTION</p> <ol style="list-style-type: none"> Do not set the PDE and PUE bits at the same time. Do not set the held value A/D conversion while PDE = 1. Do not use the pin-level self-diagnosis while PDE = 1. |
| 15 to 13 | CNVCLS[2:0] | <p>Conversion Type</p> <p>0_H: Normal A/D conversion</p> <p>1_H: Held value A/D conversion</p> <p>3_H: Self-diagnosis</p> <p>4_H: Normal A/D conversion in addition mode</p> <p>5_H: Normal A/D conversion with the MPX</p> <p>6_H: Normal A/D conversion with the MPX in addition mode</p> <p>Other than above: Setting prohibited</p> |
| 12 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

Table 30.10 ADCGnVCRj Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | |
|---|---|--|-------------|------------|--|---|--|---|---------------------------------|---|---|--|--|---|---|---|--------------------------------------|---|
| 7 | ADIE | Virtual Channel End Interrupt Enable 0: INTADCGnIx is not output at virtual channel end of virtual channel n in SGx 1: INTADCGnIx is output at virtual channel end of virtual channel n in SGx ADIE in ADCGnSGCRx is independent of ADIE in ADCGnVCRj. For details, see Section 30.4.5.1, Scan End Interrupt Request . | | | | | | | | | | | | | | | | |
| 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | | |
| 5 to 0 | GCTRL[5:0] | General Control <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>CNVCLS[2:0]</th> <th>GCTRL[5:0]</th> </tr> </thead> <tbody> <tr> <td>0_H: Normal A/D conversion</td> <td>GCTRL[4:0]: Physical channel number m*¹</td> </tr> <tr> <td>1_H: Held value A/D conversion</td> <td>Physical Channel number k of T&H supported pin 00_H: The voltage held by T&H0 is A/D converted. 01_H: The voltage held by T&H1 is A/D converted. 02_H: The voltage held by T&H2 is A/D converted. 03_H: The voltage held by T&H3 is A/D converted. 04_H: The voltage held by T&H4 is A/D converted. 05_H: The voltage held by T&H5 is A/D converted. Other than above: Setting prohibited Specify a pin for T&H listed in Table 30.6, External Input/Output Signals. Setting any of pins not supported for T&H is prohibited.</td> </tr> <tr> <td>3_H: Self-diagnosis</td> <td>00_H: AnVREFH × 0 04_H: AnVREFH × 1/4 08_H: AnVREFH × 1/2 0C_H: AnVREFH × 3/4 10_H: AnVREFH × 1 Other than above: Setting prohibited</td> </tr> <tr> <td>4_H: Normal A/D conversion in addition mode</td> <td>GCTRL[4:0]: Physical channel number m*¹ The count specified by ADDNT is applied to the number of additions.</td> </tr> <tr> <td>5_H: Normal A/D conversion (with the external MPX)</td> <td>GCTRL[4:0]: Set physical channel number m of MPX supported pin.</td> </tr> <tr> <td>6_H: Normal A/D conversion (with the external MPX in addition mode)</td> <td>GCTRL[4:0]: Set physical channel number m of MPX supported pin.</td> </tr> <tr> <td>Other than above: Setting prohibited</td> <td>—</td> </tr> </tbody> </table> | CNVCLS[2:0] | GCTRL[5:0] | 0 _H : Normal A/D conversion | GCTRL[4:0]: Physical channel number m* ¹ | 1 _H : Held value A/D conversion | Physical Channel number k of T&H supported pin 00 _H : The voltage held by T&H0 is A/D converted. 01 _H : The voltage held by T&H1 is A/D converted. 02 _H : The voltage held by T&H2 is A/D converted. 03 _H : The voltage held by T&H3 is A/D converted. 04 _H : The voltage held by T&H4 is A/D converted. 05 _H : The voltage held by T&H5 is A/D converted. Other than above: Setting prohibited Specify a pin for T&H listed in Table 30.6, External Input/Output Signals . Setting any of pins not supported for T&H is prohibited. | 3 _H : Self-diagnosis | 00 _H : AnVREFH × 0 04 _H : AnVREFH × 1/4 08 _H : AnVREFH × 1/2 0C _H : AnVREFH × 3/4 10 _H : AnVREFH × 1 Other than above: Setting prohibited | 4 _H : Normal A/D conversion in addition mode | GCTRL[4:0]: Physical channel number m* ¹ The count specified by ADDNT is applied to the number of additions. | 5 _H : Normal A/D conversion (with the external MPX) | GCTRL[4:0]: Set physical channel number m of MPX supported pin. | 6 _H : Normal A/D conversion (with the external MPX in addition mode) | GCTRL[4:0]: Set physical channel number m of MPX supported pin. | Other than above: Setting prohibited | — |
| CNVCLS[2:0] | GCTRL[5:0] | | | | | | | | | | | | | | | | | |
| 0 _H : Normal A/D conversion | GCTRL[4:0]: Physical channel number m* ¹ | | | | | | | | | | | | | | | | | |
| 1 _H : Held value A/D conversion | Physical Channel number k of T&H supported pin 00 _H : The voltage held by T&H0 is A/D converted. 01 _H : The voltage held by T&H1 is A/D converted. 02 _H : The voltage held by T&H2 is A/D converted. 03 _H : The voltage held by T&H3 is A/D converted. 04 _H : The voltage held by T&H4 is A/D converted. 05 _H : The voltage held by T&H5 is A/D converted. Other than above: Setting prohibited Specify a pin for T&H listed in Table 30.6, External Input/Output Signals . Setting any of pins not supported for T&H is prohibited. | | | | | | | | | | | | | | | | | |
| 3 _H : Self-diagnosis | 00 _H : AnVREFH × 0 04 _H : AnVREFH × 1/4 08 _H : AnVREFH × 1/2 0C _H : AnVREFH × 3/4 10 _H : AnVREFH × 1 Other than above: Setting prohibited | | | | | | | | | | | | | | | | | |
| 4 _H : Normal A/D conversion in addition mode | GCTRL[4:0]: Physical channel number m* ¹ The count specified by ADDNT is applied to the number of additions. | | | | | | | | | | | | | | | | | |
| 5 _H : Normal A/D conversion (with the external MPX) | GCTRL[4:0]: Set physical channel number m of MPX supported pin. | | | | | | | | | | | | | | | | | |
| 6 _H : Normal A/D conversion (with the external MPX in addition mode) | GCTRL[4:0]: Set physical channel number m of MPX supported pin. | | | | | | | | | | | | | | | | | |
| Other than above: Setting prohibited | — | | | | | | | | | | | | | | | | | |

Write 0 to the bits in GCTRL[5:0] which are not used.

Note 1. Specify a physical channel ANNm listed in **Table 29.7, PIC1A External Input/Output Signals**. Setting any of physical channels ANNm not incorporated is prohibited.

CAUTION

To prevent malfunctions, make ADCGnVCRj settings after making or confirming the following settings in the order shown:

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACT of all scan groups are 0 (before scan groups are started).

However, if the above are for scan groups where ADCGnVCRj.PUE and ADCGnVCRj.PDE are being used to detect wiring breaks, the ADCGnVCRj.GCTRL[5:0] bits can be set even when not all of the SGACT bits are 0 (0 indicating that conversion by the scan groups has not started yet). Only the SGACT bits for the scan groups to be set need to be 0.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.5 ADCGnDRj — Data Register j

ADCGnDRj is a 32-bit read-only register that stores the A/D conversion results corresponding to ADCGnVCRj and ADCGnVCR(j+1). As the A/D conversion results, the conversion result for ADCGnVCR(j+1) is stored in the upper bits (ADCGnDR(j+1)), and the conversion result for ADCGnVCRj is stored in the lower bits (ADCGnDRj). The format of ADCGnDRj depends on the setting of ADDNT when DFMT and CNVCLS[2:0] in ADCGnVCRj = 4H, 6H. ADCGnDRj is cleared to 0000_H when ADCGnDRj or ADCGnDIRj is read while RDCLRE is set to 1.

Access: This register can be read in 32-bit units.

Address: <ADCGn_base> + 100_H + j × 2_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | DRj+1[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DRj[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 30.11 ADCGnDRj Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 16 | DRj+1[15:0] | Stores the A/D conversion result data of DR(j+1)[15:0]. (The A/D conversion result for the channel set in ADCGnVCR(j+1) is transferred.) |
| 15 to 0 | DRj [15:0] | Stores the A/D conversion result data of DRj[15:0]. (The A/D conversion result for the channel set in ADCGnVCRj is transferred.) |

Note: j = 00, 02, 04 ... only even numbers

For signed fixed-point format (DFMT = 0)

| Number of addition | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Convert once | S | | | | | | | | | | | | | 0 | 0 | 0 |
| Convert twice | S | | | | | | | | | | | | | | 0 | 0 |
| Convert 4 timers | S | | | | | | | | | | | | | | | 0 |

↑ Position of decimal point

For signed integer format (DFMT = 1)

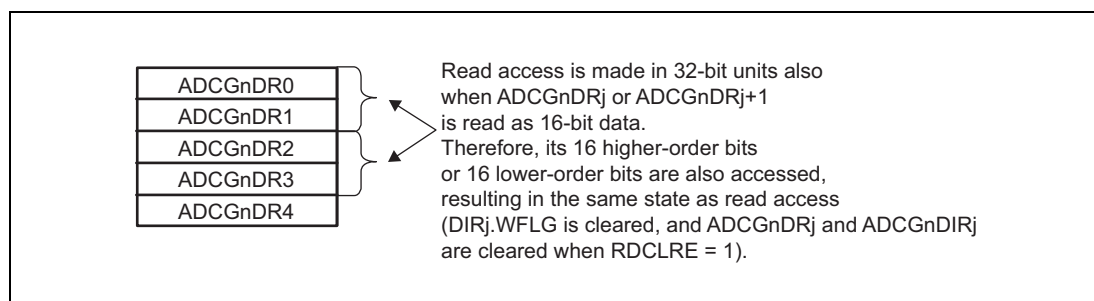
| Number of addition | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Convert once | S | S | S | S | | | | | | | | | | | | |
| Convert twice | S | S | S | | | | | | | | | | | | | |
| Convert 4 timers | S | S | | | | | | | | | | | | | | |

↗
Position of decimal point

- S : Sign bit (always 0)
- 0 : Zero extension

The format setting in ADDNT is valid when CNVCLS[2:0] = 4_H or 6_H.

If CNVCLS[2:0] is neither 4_H nor 6_H, the format is “convert once.”



NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.6 ADCGnDIRj — Data Supplementary Information Register j

ADCGnDIRj is a 32-bit read-only register that stores supplementary information on ADCGnDRj and the A/D converted value. ADCGnDIRj is provided for each virtual channel. ADCGnDIRj is cleared to 0000 0000_H when ADCGnDRj or ADCGnDIRj is read while RDCLRE is set to 1. WFLG is cleared when ADCGnDRj or ADCGnDIRj is read regardless of the RDCLRE setting. This register must always be read as 32-bit data. ADCGnDRj is read from the 16 lower-order bits.

Access: This register can be read in 32-bit units.

Address: <ADCGn_base> + 200_H + j × 4_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|------|------|----|----|----|---------|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | WFLG | PRTY | — | — | — | ID[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DRj[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 30.12 ADCGnDIRj Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 26 | Reserved | When read, the value after reset is read. |
| 25 | WFLG | Write Flag <ul style="list-style-type: none"> Setting condition An A/D converted value is stored in ADCGnDRj Clearing conditions ADCGnDRj or ADCGnDIRj is read. |
| 24 | PRTY | Parity Parity bit (even parity) for ADCGnDRj and ID[4:0] of ADCGnDIRj. |
| 23 to 21 | Reserved | When read, the value after reset is read. |

Table 30.12 ADCGnDIRj Register Contents (2/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | |
|--|---|---|-------------|---------|--|--|--|---|---------------------------------|---|---|--|---|--|--|--|--------------------------------------|---|
| 20 to 16 | ID[4:0] | Stores the ID information. | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>CNVCLS[2:0]</th> <th>ID[4:0]</th> </tr> </thead> <tbody> <tr> <td>0_H: Normal A/D conversion</td> <td>Stores the physical channel information which has actually been A/D converted. ID[4:0]: Physical channel number</td> </tr> <tr> <td>1_H: Held value A/D conversion</td> <td>Indicates the A/D converted value of the voltage held by the T&H circuit. 0_H: The voltage held by T&H0 is A/D converted. 1_H: The voltage held by T&H1 is A/D converted. 2_H: The voltage held by T&H2 is A/D converted. 3_H: The voltage held by T&H3 is A/D converted. 4_H: The voltage held by T&H4 is A/D converted. 5_H: The voltage held by T&H5 is A/D converted.</td> </tr> <tr> <td>3_H: Self-diagnosis</td> <td>Indicates changes to the self-diagnosis level. 00_H: AnVREFH × 0 04_H: AnVREFH × 1/4 08_H: AnVREFH × 1/2 0C_H: AnVREFH × 3/4 10_H: AnVREFH × 1 Other than above: Setting prohibited</td> </tr> <tr> <td>4_H: Normal A/D conversion in addition mode</td> <td>Stores the physical channel information which has actually been A/D converted. ID[4:0]: Physical channel number</td> </tr> <tr> <td>5_H: Normal A/D conversion with the MPX</td> <td>Stores the physical channel number of the MPX pin.</td> </tr> <tr> <td>6_H: Normal A/D conversion with the MPX in addition mode</td> <td>Stores the physical channel number of the MPX pin.</td> </tr> <tr> <td>Other than above: Setting prohibited</td> <td>—</td> </tr> </tbody> </table> | CNVCLS[2:0] | ID[4:0] | 0 _H : Normal A/D conversion | Stores the physical channel information which has actually been A/D converted. ID[4:0]: Physical channel number | 1 _H : Held value A/D conversion | Indicates the A/D converted value of the voltage held by the T&H circuit. 0 _H : The voltage held by T&H0 is A/D converted. 1 _H : The voltage held by T&H1 is A/D converted. 2 _H : The voltage held by T&H2 is A/D converted. 3 _H : The voltage held by T&H3 is A/D converted. 4 _H : The voltage held by T&H4 is A/D converted. 5 _H : The voltage held by T&H5 is A/D converted. | 3 _H : Self-diagnosis | Indicates changes to the self-diagnosis level. 00 _H : AnVREFH × 0 04 _H : AnVREFH × 1/4 08 _H : AnVREFH × 1/2 0C _H : AnVREFH × 3/4 10 _H : AnVREFH × 1 Other than above: Setting prohibited | 4 _H : Normal A/D conversion in addition mode | Stores the physical channel information which has actually been A/D converted. ID[4:0]: Physical channel number | 5 _H : Normal A/D conversion with the MPX | Stores the physical channel number of the MPX pin. | 6 _H : Normal A/D conversion with the MPX in addition mode | Stores the physical channel number of the MPX pin. | Other than above: Setting prohibited | — |
| CNVCLS[2:0] | ID[4:0] | | | | | | | | | | | | | | | | | |
| 0 _H : Normal A/D conversion | Stores the physical channel information which has actually been A/D converted. ID[4:0]: Physical channel number | | | | | | | | | | | | | | | | | |
| 1 _H : Held value A/D conversion | Indicates the A/D converted value of the voltage held by the T&H circuit. 0 _H : The voltage held by T&H0 is A/D converted. 1 _H : The voltage held by T&H1 is A/D converted. 2 _H : The voltage held by T&H2 is A/D converted. 3 _H : The voltage held by T&H3 is A/D converted. 4 _H : The voltage held by T&H4 is A/D converted. 5 _H : The voltage held by T&H5 is A/D converted. | | | | | | | | | | | | | | | | | |
| 3 _H : Self-diagnosis | Indicates changes to the self-diagnosis level. 00 _H : AnVREFH × 0 04 _H : AnVREFH × 1/4 08 _H : AnVREFH × 1/2 0C _H : AnVREFH × 3/4 10 _H : AnVREFH × 1 Other than above: Setting prohibited | | | | | | | | | | | | | | | | | |
| 4 _H : Normal A/D conversion in addition mode | Stores the physical channel information which has actually been A/D converted. ID[4:0]: Physical channel number | | | | | | | | | | | | | | | | | |
| 5 _H : Normal A/D conversion with the MPX | Stores the physical channel number of the MPX pin. | | | | | | | | | | | | | | | | | |
| 6 _H : Normal A/D conversion with the MPX in addition mode | Stores the physical channel number of the MPX pin. | | | | | | | | | | | | | | | | | |
| Other than above: Setting prohibited | — | | | | | | | | | | | | | | | | | |
| 15 to 0 | DRj[15:0] | Data Register The same data as that in the ADCGnDRj register is stored as an A/D conversion result. | | | | | | | | | | | | | | | | |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.7 ADCGnSMPCR — A/D Conversion Time Control Register

ADCGnSMPCR is a 16-bit readable/writable register that controls the A/D conversion time.

ADCGn supports 1- μ s conversion and 11.3- μ s conversion.

Access: This register can be read/written in 16-bit units.

Address: <ADCGn_base> + 340_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | SMPCR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.13 ADCGnSMPCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 15 to 0 | SMPCR[15:0] | Sets the A/D conversion time. 0000 _H : 1- μ s conversion 90CC _H : 11.3- μ s conversion Settings other than the above are prohibited. |

CAUTION

For 1- μ s conversion, set the CLK_ADC frequency to 40 MHz.
If you wish to use 11.3- μ s conversion, be sure to set the CLK_ADC frequency to 20 MHz.

Settings other than the above are prohibited. Use register CKSC8C to set the CLK_ADC frequency. For details, see **Section 12.3.11, CKSC8C — Clock Selector 8 Control Register**.

CAUTION

If ADCGnSMPCR is changed, the unit for counting of the wait time to be inserted when waiting for MPX will change accordingly. For details, see **Section 30.3.12, ADCGnMPXOWR — MPX Optional Wait Register**.

To prevent malfunctions, make ADCGnSMPCR settings after making or confirming the following settings in the order shown:

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACT of all scan groups are 0 (before scan groups are started).

30.3.8 ADCGnADHALTR — A/D Halt Register

ADCGnADHALTR is an 8-bit write-only register that halts the ADC. The register bits are always read as 0.

Access: This register can be written only in 8-bit units.

Address: <ADCGn_base> + 380_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|------|
| | — | — | — | — | — | — | — | HALT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.14 ADCGnADHALTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | HALT | All scan group conversions and A/D timers are halted forcibly. 0: No function (Writing 0 is ignored.) 1: Halted |

30.3.9 ADCGnADCR1 — A/D Control Register 1

ADCGnADCR1 is an 8-bit readable/writable register for ADC common control.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 384_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|-------------|-----|
| | — | — | — | — | — | — | SUSMTD[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 30.15 ADCGnADCR1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | SUSMTD [1:0] | <p>Suspend Method</p> <p>These bits select the suspend method used when a higher-priority scan group interrupts a lower-priority scan group.</p> <p>Synchronous suspend: If a request from a higher-priority SG is made while a lower-priority SG is being processed, processing for the lower-priority SG is suspended after the ongoing virtual channel processing is completed, and then processing for the higher-priority SG is executed. After processing for the higher-priority SG is completed, the suspended virtual channel processing for the lower-priority SG is resumed.</p> <p>Asynchronous suspend: If a request from a higher-priority SG is made while a lower-priority SG is being processed, the ongoing virtual channel processing is immediately suspended, and then processing for the higher-priority SG is executed. After processing for the higher-priority SG is completed, the suspended virtual channel processing for the lower-priority SG is resumed.</p> <ul style="list-style-type: none"> When ADCGnTHACR.HLDCTE = 0 and ADCGnTHBCR.HLDCTE = 0 <ul style="list-style-type: none"> 0_H: Synchronous suspend 1_H: Asynchronous suspend when a higher-priority SG interrupts SG0 Synchronous suspend when a higher-priority SG interrupts a lower-priority SG (except for SG0) 2_H: Asynchronous suspend 3_H: Setting prohibited When ADCGnTHACR.HLDCTE = 1 or ADCGnTHBCR.HLDCTE = 1 <ul style="list-style-type: none"> 2_H: Asynchronous suspend Other than above: Setting prohibited |

For details, see **Figure 30.13, Example of Synchronous Suspend and Resume Operation** and **Figure 30.14, Example of Asynchronous Suspend and Resume Operation**.

CAUTION

- To prevent malfunctions, make ADCGnADCR1 settings after making or confirming the following settings in the order shown:
 - HLDTTE of T&H group A and B are 0.
 - ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
 - SGACT of all scan groups are 0 (before scan groups are started).
- When SGx hold trigger is selected for the trigger input to SGx, only asynchronous suspend can be specified (SUMTD[1:0] = 2_H).

30.3.10 ADCGnMPXCURCR — MPX Current Control Register

ADCGnMPXCURCR is a register that controls the ADCGnMPXCURR format.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 388_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|--------------|-----|-----|-----|
| | — | — | — | — | MSKCFMT[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 30.16 ADCGnMPXCURCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | MSKCFMT[3:0] | MSKC Format Specification Specifies the format for MSKC[15:0] of ADCGnMPXCURR MSKCFMT[3] 0: MSKC[15:12] = 0000 1: MSKC[15:12] = 1111 MSKCFMT[2] 0: MSKC[11:8] = 0000 1: MSKC[11:8] = 1111 MSKCFMT[1] 0: MSKC[7:4] = 0000 1: MSKC[7:4] = 1111 MSKCFMT[0] 0: MSKC[3:0] = 0000 1: MSKC[3:0] = 1111 |

CAUTION

To prevent malfunctions, make ADCGnMPXCURCR settings after making or confirming the following settings in the order shown:

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACTION of all scan groups are 0 (before scan groups are started).

30.3.11 ADCGnMPXCURR — MPX Current Register

ADCGnMPXCURR is a 32-bit read-only register that stores the MPX value for an external analog multiplexer.

Access: This register can be read in 32-bit units.

Address: <ADCGn_base> + 38C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|----|----|----|----|----|----|----|----|----|----|-------------|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | MSKC[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | | | | | | | | | | | MPXCUR[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 30.17 ADCGnMPXCURR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 16 | MSKC[15:0] | Mask Control The format depends on the MSKCFMT[3:0] setting of ADCGnMPXCURCR. For details, see Section 30.3.10, ADCGnMPXCURCR — MPX Current Control Register . |
| 15 to 5 | Reserved | When read, the value after reset is read. |
| 4 to 0 | MPXCUR[4:0] | Current MPX value When a virtual channel for which CNVCLS[2:0] in ADCGnVCRj is set to 5 _H or 6 _H is started, GCTRL[4:0] in ADCGnVCRj is transferred to MPXCUR[4:0]. At this time, an interrupt request to the INTC or a DMA transfer request is generated. The DMAC transfers ADCGnMPXCURR to PSRn of the I/O port, enabling the MPX value to be sent to an external analog multiplexer. When PSRn is used, transfer the MPX value as a 32-bit value. This enables rewriting of only the necessary ports using the format control in MSKC[15:0]. For details, see Section 30.4.1.5, Example of Using an External Analog Multiplexer (Port Output) . |

CAUTION

For a DMA transfer to the PSRn register, the MSKC bit is specified in 4-bit units (register ADCGnMPXCURCR). When MSKCFMT[1]=1(MSKC[7:4]=1111B), the Pn[7:4] bits are updated. Do not use the 3 higher-order bits of the Pn register as output port since they are overwritten unexpectedly.

NOTE

For the indices of ADCG, see **Section 30.1.1, Number of Units**.

30.3.12 ADCGnMPXOWR — MPX Optional Wait Register

ADCGnMPXOWR is a register that specifies the wait time to be inserted for an external analog multiplexer.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 390_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------------|-----|-----|-----|
| | — | — | — | — | MPXOW[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 30.18 ADCGnMPXOWR Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 7 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | MPXOW[3:0] | <p>MPX Optional Wait</p> <p>These bits specify the wait time to be inserted before A/D conversion is started after a virtual channel for which CNVCLS[2:0] in ADCGnVCRj is 5_H or 6_H is started.</p> <p>1_H: A/D conversion time × 1 2_H: A/D conversion time × 2 3_H: A/D conversion time × 3 4_H: A/D conversion time × 4 5_H: A/D conversion time × 5 6_H: A/D conversion time × 6 7_H: A/D conversion time × 7 8_H: A/D conversion time × 8 9_H: A/D conversion time × 9 A_H: A/D conversion time × 10 B_H to F_H: Setting prohibited</p> <p>For the A/D conversion time, see Section 30.3.7, ADCGnSMPCR — A/D Conversion Time Control Register.</p> |

CAUTION

To prevent malfunctions, make ADCGnMPXOWR settings after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACT of all scan groups are 0 (before scan groups are started).

30.3.13 ADCGnADCR2 — A/D Control Register 2

ADCGnADCR2 is an 8-bit readable/writable register for ADCG common control.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 398_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|------|---|---|---|-------|
| | — | — | — | DFMT | — | — | — | ADDNT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R | R | R | R/W |

Table 30.19 ADCGnADCR2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | DFMT | Data Format 0: Signed fixed-point format 1: Signed integer format This bit specifies the format of data to be transferred to ADCGnDRj. For details of data format, see Section 30.3.5, ADCGnDRj — Data Register j . |
| 3 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | ADDNT | Addition Count Select 0: Add twice 1: Add 4 times This bit is valid only when CNVCLS[2:0] is 4 _H ,6 _H |

CAUTION

To prevent malfunctions, make ADCGnADCR2 settings after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACT of all scan groups are 0 (before scan groups are started).

30.3.14 ADCGnADENDPz — A/D Conversion Monitor Virtual Channel Pointer z

ADCGnADENDPz is an 8-bit readable/writable register which selects the virtual channel which outputs the A/D conversion timing to ADCGnCNVx.

Access: This register can be read/written in 8-bit units.

Address: ADCGnADENDP0: <ADCGn_base> + 3A0_H
 ADCGnADENDP1: <ADCGn_base> + 3A4_H
 ADCGnADENDP2: <ADCGn_base> + 3A8_H
 ADCGnADENDP3: <ADCGn_base> + 3AC_H
 ADCGnADENDP4: <ADCGn_base> + 3B0_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|-----------|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | ENDP[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.20 ADCGnADENDPz Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | ENDP[5:0] | A/D Conversion Monitor Virtual Channel Pointer When the virtual channel selected by ADCGnADENDPz is started, a high level is output from the ADCGnCNVz pin. When the virtual channel selected by ADCGnADENDPz ends, a low level is output. |

CAUTION

To prevent malfunctions, make ADCGnADENDPz settings after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACT of all scan groups are 0 (before scan groups are started).

30.3.15 ADCGnTHSMPSTCR — T&H Sampling Start Control Register

ADCGnTHSMPSTCR is an 8-bit write-only register that control starting of all T&H sampling. The register bits are always read as 0.

Access: This register can be written in 8-bit units.

Address: <ADCGn_base> + 400_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | SMPST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.21 ADCGnTHSMPSTCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SMPST | Starts T&H sampling. 0: No function. (Writing 0 is ignored.) 1: Sampling is started. Conditions for T&Hk to transit to the sampling state: 1 is written to SMPST while THkE = 1 At completion of A/D conversion of the value held by T&Hz while THkE = 1 and ASMPMSK = 0 |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.16 ADCGnTHSTPCR — T&H Stop Control Register

ADCGnTHSTPCR is an 8-bit write-only register that controls stopping of all T&H. The register bits are always read as 0.

Access: This register can be written in 8-bit units.

Address: <ADCGn_base> + 404_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | THSTP |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.22 ADCGnTHSTPCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | THSTP | Stops T&H. 0: No function. (Writing 0 is ignored.) 1: T&H is stopped. |

CAUTION

Set THSTP after all scan groups are stopped by ADCGnADHALTR.HALT.

After T&H are forcibly terminated by THSTP, set all THzE (z = 0 to 5) of ADCGnTHER to 0 to prevent resampling by auto sampling function (ADCGnTHCR.ASMPMASK).

30.3.17 ADCGnTHCR — T&H Control Register

ADCG0THCR in an 8-bit readable/writable register that controls T&H.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 408_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | ASMPMSK |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 30.23 ADCGnTHCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | ASMPMSK | Controls whether to start auto sampling of the T&H when A/D conversion of the relevant held value is complete. 0: Automatic sampling is performed. 1: Automatic sampling is not performed. |

CAUTION

To prevent malfunctions, make ADCGnTHCR settings after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) SGACT of the scan group x specified by SGS[1:0] of T&H group A and B are 0 (before scan groups are started).
- (3) THkE of all T&H are 0 (All T&H are stopped).

30.3.18 ADCGnTHAHLDDSTCR — T&H Group A Hold Start Control Register

ADCGnTHAHLDDSTCR is an 8-bit write-only register that controls the start of the hold for T&H group A. The register bits are always read as 0.

Access: This register can be written in 8-bit units.

Address: <ADCGn_base> + 410_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-------|
| | — | — | — | — | — | — | — | HLDST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.24 ADCGnTHAHLDDSTCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | HLDST | Starts the hold of T&H group A. 0: No function (Writing 0 is ignored.) 1: The hold is started. |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.19 ADCGnTHBHLDDSTCR — T&H Group B Hold Start Control Register

ADCGnTHBHLDDSTCR is an 8-bit write-only register that controls the start of the hold for T&H group B. The register bits are always read as 0.

Access: This register can be written in 8-bit units.

Address: <ADCGn_base> + 414_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-------|
| | — | — | — | — | — | — | — | HLDST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.25 ADCGnTHAHLDDSTCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | HLDST | Starts the hold of T&H group B. 0: No function (Writing 0 is ignored.) 1: The hold is started. |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.20 ADCGnTHACR — T&H Group A Control Register

ADCGnTHACR is an 8-bit readable/writable register that controls T&H group A.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 420_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|--------|--------|---|---|----------|-----|
| | — | — | HLDCTE | HLDTTE | — | — | SGS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R/W | R/W |

Table 30.26 ADCGnTHACR Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|----------|--|--------|--------|----------|---|---|--|---|---|---------------------|---|---|---|---|---|---|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 5 | HLDCTE | | | | | | | | | | | | | | | | |
| 4 | HLDTTE | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>HLDCTE</th> <th>HLDTTE</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>The hold trigger of T&H group A is disabled.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Setting prohibited.</td> </tr> <tr> <td>1</td> <td>0</td> <td>The software trigger by the ADCGnTHAHLDTSTCR.HLDST is selected as the hold trigger for the SG selected in SGS[1:0]. Hardware trigger is disabled. Note</td> </tr> <tr> <td>1</td> <td>1</td> <td>SGx_TRG hardware trigger is selected as the hold trigger for the SG selected in SGS[1:0]. Note</td> </tr> </tbody> </table> | HLDCTE | HLDTTE | Function | 0 | 0 | The hold trigger of T&H group A is disabled. | 0 | 1 | Setting prohibited. | 1 | 0 | The software trigger by the ADCGnTHAHLDTSTCR.HLDST is selected as the hold trigger for the SG selected in SGS[1:0]. Hardware trigger is disabled. Note | 1 | 1 | SGx_TRG hardware trigger is selected as the hold trigger for the SG selected in SGS[1:0]. Note |
| HLDCTE | HLDTTE | Function | | | | | | | | | | | | | | | |
| 0 | 0 | The hold trigger of T&H group A is disabled. | | | | | | | | | | | | | | | |
| 0 | 1 | Setting prohibited. | | | | | | | | | | | | | | | |
| 1 | 0 | The software trigger by the ADCGnTHAHLDTSTCR.HLDST is selected as the hold trigger for the SG selected in SGS[1:0]. Hardware trigger is disabled. Note | | | | | | | | | | | | | | | |
| 1 | 1 | SGx_TRG hardware trigger is selected as the hold trigger for the SG selected in SGS[1:0]. Note | | | | | | | | | | | | | | | |
| | | <p>Note: A/D conversion is forcibly suspended during the period from hold trigger to hold completion. SUSMTD[1:0] should be 2_H. (Setting of SUSMTD[1:0] = 0_H or 1_H is prohibited.) The setting of TRGMD of the scan group selected in SGS[1:0] should be 1_H (Setting of TRGMD = 0_H, 2_H, or 3_H is prohibited.)</p> <p>When modifying HLDTE from 1 to 0 (SGx_TRG hardware trigger is disabled) when HLDCTE = 1 and during operation, follow the procedure below: (1) Change HLDTE from 1 to 0 (2) Change TRGMD of the scan group x selected in SGS[1:0] from 1 to 0 When modifying HLDTE from 0 to 1 (SGx_TRG hardware trigger is selected): (1) Change TRGMD of the scan group x selected in SGS[1:0] from 0 to 1 (2) Change HLDTE from 0 to 1.</p> | | | | | | | | | | | | | | | |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 1, 0 | SGS[1:0] | <p>Scan Group Select Selects the scan group for T&H group A.</p> <p>0_H: SG1 1_H: SG2 2_H: SG3 3_H: SG4</p> <p>For details, see Section 30.4.1.3, Simultaneous Track and Hold Operation (THC Control).</p> <p>CAUTION</p> <p>This bit is enabled when HLDCTE = 1. Designate a scan group which include the virtual channel subject to T&H conversion.</p> | | | | | | | | | | | | | | | |

CAUTION

To prevent malfunctions, make settings for ADCGnTHACR.HLDCTE and ADCG0THACR.SGS[1:0] after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACTION of all scan groups are 0 (before scan groups are started).
- (4) THkE of all T&H are 0 (all T&H are stopped).

Do not set the SGS[1:0] bit of ADCGnTHACR and ADCGnTHBCR to the same scan group when both T&H group A and B are used.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.21 ADCGnTHBCR — T&H Group B Control Register

ADCGnTHBCR is an 8-bit readable/writable register that controls T&H group B.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 424_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|--------|--------|---|---|----------|-----|
| | — | — | HLDCTE | HLDTTE | — | — | SGS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R/W | R/W |

Table 30.27 ADCGnTHBCR Register Contents

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|----------|---|--------|--------|----------|---|---|--|---|---|---------------------|---|---|---|---|---|---|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 5 | HLDCTE | | | | | | | | | | | | | | | | |
| 4 | HLDTTE | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>HLDCTE</th> <th>HLDTTE</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>The hold trigger of T&H group B is disabled.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Setting prohibited.</td> </tr> <tr> <td>1</td> <td>0</td> <td>The software trigger by the ADCGnTHBHLDTSTCR.HLDST is selected as the hold trigger for the SG selected in SGS[1:0]. Hardware trigger is disabled. Note</td> </tr> <tr> <td>1</td> <td>1</td> <td>SGx_TRG hardware trigger is selected as the hold trigger for SG selected in SGS[1:0]. Note</td> </tr> </tbody> </table> | HLDCTE | HLDTTE | Function | 0 | 0 | The hold trigger of T&H group B is disabled. | 0 | 1 | Setting prohibited. | 1 | 0 | The software trigger by the ADCGnTHBHLDTSTCR.HLDST is selected as the hold trigger for the SG selected in SGS[1:0]. Hardware trigger is disabled. Note | 1 | 1 | SGx_TRG hardware trigger is selected as the hold trigger for SG selected in SGS[1:0]. Note |
| HLDCTE | HLDTTE | Function | | | | | | | | | | | | | | | |
| 0 | 0 | The hold trigger of T&H group B is disabled. | | | | | | | | | | | | | | | |
| 0 | 1 | Setting prohibited. | | | | | | | | | | | | | | | |
| 1 | 0 | The software trigger by the ADCGnTHBHLDTSTCR.HLDST is selected as the hold trigger for the SG selected in SGS[1:0]. Hardware trigger is disabled. Note | | | | | | | | | | | | | | | |
| 1 | 1 | SGx_TRG hardware trigger is selected as the hold trigger for SG selected in SGS[1:0]. Note | | | | | | | | | | | | | | | |
| | | <p>Note: A/D conversion is forcibly suspended during the period from hold trigger to hold completion. SUSMTD[1:0] should be 2_H. (Setting of SUSMTD[1:0] = 0_H or 1_H is prohibited.) The setting of TRGMD of the scan group selected in SGS[1:0] should be 1H (Setting of TRGMD = 0_H, 2_H, or 3_H is prohibited.)</p> <p>When modifying HLDTE from 1 to 0 (SGx_TRG hardware trigger is disabled) when HLDCTE = 1 and during operation, follow the procedure below: (1) Change HLDTE from 1 to 0 (2) Change TRGMD of the scan group x selected in SGS[1:0] from 1 to 0 When modifying HLDTE from 0 to 1 (SGx_TRG hardware trigger is selected): (1) Change TRGMD of the scan group x selected in SGS[1:0] from 0 to 1 (2) Change HLDTE from 0 to 1.</p> | | | | | | | | | | | | | | | |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 1, 0 | SGS[1:0] | <p>Scan Group Select Selects the scan group for T&H group B.</p> <p>0_H: SG1 1_H: SG2 2_H: SG3 3_H: SG4</p> <p>For details, see Section 30.4.1.3, Simultaneous Track and Hold Operation (THC Control).</p> <p>CAUTION</p> <p>This bit is enabled when HLDCTE = 1. Designate a scan group which include the virtual channel subject to T&H conversion.</p> | | | | | | | | | | | | | | | |

CAUTION

To prevent malfunctions, make settings for ADCGnTHBCR.HLDCTE and ADCG0THBCR.SGS[1:0] after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACT of all scan groups are 0 (before scan groups are started).
- (4) THkE of all T&H are 0 (all T&H are stopped).

Do not set the SGS[1:0] bit of ADCGnTHACR and ADCGnTHBCR to the same scan group when both T&H group A and B are used.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.22 ADCGnTHER — T&H Enable Register

ADCGnTHER is an 8-bit readable/writable register that controls enabling and disabling of each T&H.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 430_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|------|------|------|------|------|------|
| | — | — | TH5E | TH4E | TH3E | TH2E | TH1E | TH0E |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.28 ADCGnTHER Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | THkE | Enables or disables track and hold operation of T&Hk circuit. 0: Disabled 1: Enabled For details, see Section 30.4.1.3, Simultaneous Track and Hold Operation (THC Control) . |

CAUTION

To prevent malfunctions, make ADCGnTHER settings after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) SGACT of the scan group x specified in SGS[1:0] of the T&H group A and B are 0 (before scan groups are started).

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.23 ADCGnTHGSR — T&H Group Select Register

ADCGnTHGSR is a 16-bit readable/writable register that selects the T&H group of each T&H.

Access: This register can be read/written in 16-bit units.

Address: <ADCGn_base> + 434_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|-------|---|-------|---|-------|---|-------|---|-------|---|-------|
| | — | — | — | — | — | TH5GS | — | TH4GS | — | TH3GS | — | TH2GS | — | TH1GS | — | TH0GS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/W | R | R/W | R | R/W | R | R/W | R | R/W | R | R/W |

Table 30.29 ADCGnTHGSR Register Contents

| Bit Position | Bit Name | Function |
|-------------------------|----------|--|
| 15 to 11, 9, 7, 5, 3, 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 10, 8, 6, 4, 2, 0 | THkGS | Selects the T&H group for T&Hk. 0: T&H group A is selected. 1: T&H group B is selected. For details, see Section 30.4.1.3, Simultaneous Track and Hold Operation (THC Control) . |

CAUTION

To prevent malfunctions, make settings for ADCGnTHGSR after making or confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) SGACT of the scan group x specified in SGS[1:0] of the T&H group A and B are 0 (before scan groups are started).
- (3) THkE of all T&H are 0 (all T&H are stopped).

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.24 ADCGnSFTCR — Safety Control Register

ADCGnSFTCR is an 8-bit readable/writable register for safety control.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 3C0_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|--------|-------|-------|------|-------|
| | — | — | — | RDCLRE | ULEIE | OWEIE | PEIE | IDEIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 30.30 ADCGnSFTCR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | RDCLRE | Read and Clear Enable/Disable This bit sets whether to clear (ALL 0) the ADCGnDRj and ADCGnDIRj registers when reading the ADCGnDRj or ADCGnDIRj. 0: Do not perform read and clear 1: Perform read and clear CAUTION ADCGnDIRj.WFLG is cleared by reading ADCGnDRj or ADCGnDIRj regardless of the RDCLRE setting. |
| 3 | ULEIE | Enables or disables upper-limit/lower-limit error interrupt. 0: Disabled 1: Enabled |
| 2 | OWEIE | Enables or disables overwrite error interrupt. 0: Disabled 1: Enabled |
| 1 | PEIE | Enables or disables parity error interrupt. 0: Disabled 1: Enabled |
| 0 | IDEIE | Enables or disables ID error interrupt. 0: Disabled 1: Enabled |

CAUTION

To prevent malfunctions, make ADCGnSFTCR settings after making of confirming the following settings.

- (1) HLDTE of T&H group A and B are 0.
- (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- (3) SGACTION of all scan groups are 0 (before the scan group is started).

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.25 ADCGnTDCR — Pin Level Self-Diagnostic Control Register

ADCGnTDCR is an 8-bit readable/writable register that controls the pin level self-diagnosis.

Access: This register can be read/written in 8-bit units.

Address: <ADCGn_base> + 3C4_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----|---|---|---|---|---|-----------|-----|
| | TDE | — | — | — | — | — | TDLV[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R | R | R/W | R/W |

Table 30.31 ADCGnTDCR Register Contents

| Bit Position | Bit Name | Function |
|---|-----------|---|
| 7 | TDE | <p>Enables or disables pin level self-diagnosis.</p> <p>0: Pin level self-diagnosis is disabled.</p> <p>1: Pin level self-diagnosis is enabled.</p> <p>When TDE is set to 1, all analog pins are disconnected from the input buffer. When TDE is set to 0, all analog pins are connected to the input buffer. When TDE is set to 1, the voltage is fixed to the level specified by TDLV[1:0]. Performing A/D conversion in this state and checking the A/D converted value allows diagnosis of the path from an analog pin to the ADCG.</p> |
| <p>CAUTION</p> <ol style="list-style-type: none"> When TDE = 1, be sure to set both the PUE and PDE bits of ADCGnVCRj to 0. Set ADCGnTDCR.TDE to 0 before setting the PUE and PDE bits of ADCGnVCRj to 1. Similarly, when setting ADCGnTDCR.TDE to 1, set the PUE and PDE bits of ADCGnVCRj targeted for A/D conversion to 0 while ADCGnTDCR.TDE = 1 (the recommended setting is 0 for the PUE and PDE bits of all ADCGnVCRj registers). | | |
| 6 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TDLV[1:0] | <p>Sets pin level self-diagnosis voltage level.</p> <p>0: Apply AnVSS to even-numbered physical channel groups and AnVCC to odd-numbered physical channel groups.</p> <p>1: Apply AnVCC to even-numbered physical channel groups and AnVSS to odd-numbered physical channel groups.</p> <p>2: Apply AnVSS to even-numbered physical channel groups and 1/2 × AnVCC to odd-numbered physical channel groups.</p> <p>3: Apply 1/2 × AnVCC to even-numbered physical channel groups and 1/2 × AnVSS to odd-numbered physical channel groups.</p> |

CAUTION

To prevent malfunctions, make ADCGnTDCR settings after making or confirming the following settings.

- HLDTE of T&H group A and B are 0.
- ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
- SGACT of all scan groups are 0 (before scan groups are started).

When **Section 30.4.4.5, Pin-Level Self-Diagnosis of T&H Paths** is used, however, settings of ADCGnTDCR of the scan group in progress is possible only if the procedure is followed correctly.

30.3.26 ADcGnODcR — Wiring Break Detection Control Register

ADcGnODcR is a 32-bit readable/writable register that controls wiring-break detection.

Access: This register can be read/written in 32-bit units.

Address: <ADcGn_base> + 3C8_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | ODDE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|-----|---|-----------|-----|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | ODE | — | ODPW[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.32 ADcGnODcR Register Contents

| Bit Position | Bit Name | Function |
|--|-----------|---|
| 31 | ODDE | Enables or disables diagnosis of wiring-break detection. 0: Wiring-break detection is not diagnosed. 1: Wiring-break detection is diagnosed. When the ODDE bit is set to 1, diagnosis of wiring-break detection is enabled for all analog pins. |
| CAUTION | | |
| Do not set the ODE and ODDE bits at the same time. | | |
| 30 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 | ODE | Sets the mode for wiring-break detection. 0: Wiring-break detection mode 2 1: Wiring-break detection mode 1 For details, see Section 30.4.4.3, Wiring-Break Detection . |
| 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | ODPW[5:0] | Sets the width of pull-down pulse to be generated in wiring-break detection mode 1. 04 _H : 1 clock 05 _H : 2 clocks : 13 _H : 16 clocks 14 _H : 17 clocks A value larger than 03 _H and less than 15 _H is required. |

CAUTIONS

1. To prevent malfunctions, make ADCGnODCR settings after making or confirming the following settings.
 - (1) HLDTE of T&H group A and B are 0.
 - (2) ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
 - (3) SGACT of all scan groups are 0 (before scan groups are started).
 2. Do not perform A/D conversion of the held value (ADCGnVCRj.CNVCLS = 1_H) with wiring-break detection enabled.
 3. When Wiring-break Detection mode 1 (ADCGnODCR.ODE = 1_H) is used, to prevent malfunctions, use the wiring-break detection function after setting all THkE (k = 0 to 5) of ADCGnTHER to 0 or stopping all T&H by THSTPCR.THSTP.
-

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.27 ADOPDIGn — Wiring-break Detection Pin Setting Register n (n = 0, 1)

ADOPDIGn is a register that controls the pull-down/pull-up switch for I/O buffer of each analog input.

Access: This register can be read/written in 32-bit units.

Address: ADOPDIG0: FFF9 1800_H
ADOPDIG1: FFF9 2800_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | ADOPDI Gn11 | ADOPDI Gn10 | ADOPDI Gn09 | ADOPDI Gn08 | ADOPDI Gn07 | ADOPDI Gn06 | ADOPDI Gn05 | ADOPDI Gn04 | ADOPDI Gn03 | ADOPDI Gn02 | ADOPDI Gn01 | ADOPDI Gn00 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.33 ADOPDIGn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 to 0 | ADOPDIGnm | Controls the pull-down/pull-up settings for the wiring-break detection pin. Specify an analog pin for wiring-break detection. 0: Wiring-break Detection is not performed. 1: Wiring-break Detection is performed. When Wiring-break Detection mode 1 is used, wiring-break detection for the analog pin which has this bit set to 1 is performed. When Wiring-break Detection mode 2 is used, the pull-up/pull-down setting (ADCGnVCRj.PDE/ADCGnVCRj.PUE) of the analog pin which has this bit set to 1 is enabled. |

CAUTIONS

- To prevent malfunctions, make ADOPDIGn settings after making or confirming the following settings.
 - HLDTTE of T&H group A and B are 0.
 - ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
 - SGACT of all scan groups are 0 (before scan groups are started).
- Do not perform A/D conversion of the held value (ADCGnVCRj.CNVCLS = 1_H) with wiring-break detection enabled.
- When Wiring-break Detection mode 1 (ADCGnODCR.ODE = 1_H) is used, to prevent malfunctions, use the wiring-break detection function after setting all THkE (k = 0 to 5) of ADCGnTHER to 0 or stopping all T&H by THSTPCR.THSTP.

30.3.28 ADCGnULLMTBR0 to 2 — Upper-limit/Lower-limit Table Registers 0 to 2

ADCGnULLMTBR0-2 are 32-bit readable/writable registers that set the upper-limit and lower-limit values of an A/D converted value. Specify any of ADCGnULLMTBR0 to 2 by ULS[1:0] in ADCGnULLMSRx.

Access: This register can be read/written in 32-bit units.

Address: ADCGmULLMTBR0: <ADCGn_base> + 3CC_H
 ADCGmULLMTBR1: <ADCGn_base> + 3D0_H
 ADCGmULLMTBR2: <ADCGn_base> + 3D4_H

Value after reset: 7FFE 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| | ULMTB[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LLMTB[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |

Table 30.34 ADCGnULLMTBR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 16 | ULMTB[15:0] | <p>Upper Limit Table</p> <p>These bits specify the upper-limit value of an A/D converted value. ULE (upper limit/lower limit error) is set when the following condition is met.</p> <p>ULMTB[15:0] < A/D converted value</p> <p>The ULMTB[15:0] format is the signed fixed-point format regardless of the format of ADCGnDRj. If the signed integer format is selected for the ADCGnDRj format, the ADCGnDRj format is converted to the signed fixed-point format, and then the values are compared. Note that ULMTB[15] and ULMTB[0] are always fixed to 0.</p> |
| 15 to 0 | LLMTB[15:0] | <p>Lower Limit Table</p> <p>These bits specify the lower-limit value of an A/D converted value. ULE (upper limit/lower limit error) is set when the following condition is met.</p> <p>LLMTB[15:0] > A/D converted value</p> <p>The LLMTB[15:0] format is the signed fixed-point format regardless of the format of ADCGnDRj. If the signed integer format is selected for the ADCGnDRj format, the ADCGnDRj format is converted to the signed fixed-point format, and then the values are compared. Note that LLMTB[15] and LLMTB[0] are always fixed to 0.</p> |

CAUTION

- To prevent malfunctions, make settings for ADCGnULLMTBR0 to 2 after making or confirming the following settings.
 - HLDTE of T&H group A and B are 0.
 - ADSTARTE of all scan groups are 0 and TRGMD of all scan groups are 0_H.
 - SGACT of all scan groups are 0 (before scan groups are started).
- The upper table (ULMTB[15:0]) must be set greater than the lower table (LLMTB[15:0]).

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.29 ADCGnECR — Error Clear Register

ADCGnECR is an 8-bit write-only register that controls error clear. The register bits are always read as 0.

Access: This register can be written only in 8-bit units.

Address: <ADCGn_base> + 3D8_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|------|------|-----|------|
| | — | — | — | — | ULEC | OWEC | PEC | IDEC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | W | W | W | W |

Table 30.35 ADCGnECR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 4 | Reserved | When writing, write the value after reset. |
| 3 | ULEC | Upper Limit/Lower Limit Error Clear Clears the upper limit or lower limit errors (ADCGnULER.ULE) and the upper limit or lower limit error capture (ADCGnULER.ULECAP[5:0]). 0: Not cleared 1: Cleared |
| 2 | OWEC | Overwrite Error Clear Clears over write errors (ADCGnOWER.OWE) and over write error capture (ADCGnOWER.OWECAP[5:0]). 0: Not cleared 1: Cleared |
| 1 | PEC | Parity Error Clear Clears parity errors (ADCGnPER.PE) and parity error capture (ADCGnPER.PECAP[5:0]). 0: Not cleared 1: Cleared |
| 0 | IDEC | ID Error Clear Clears ID errors (ADCGnIER.IDE) and ID error capture (ADCGnIER.IDECAP[5:0]). 0: Not cleared 1: Cleared |

30.3.30 ADCGnULER — Upper-limit/Lower-limit Error Register

ADCGnULER is an 8-bit read-only register that indicates upper limit/lower limit errors.

Access: This register can be read only in 8-bit units.

Address: <ADCGn_base> + 3DC_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------------|-----|---|-------------|---|---|---|---|---|--|
| | ULE | — | ULECAP[5:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | |

Table 30.36 ADCGnULER Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 7 | ULE | Upper Limit/Lower Limit Error 0: No error 1: An error is present. Setting condition The A/D converted value exceeds the range of the specified upper limit/lower limit table Clearing condition A value of 1 is written to ULEC. |
| 6 | Reserved | When read, the value after reset is read. |
| 5 to 0 | ULECAP[5:0] | Upper Limit/Lower Limit Error Capture The virtual channel at the time when an upper limit/lower limit error occurred is captured. Capturing condition ULE = 0 and the A/D converted value exceeds the range of the specified upper limit/lower limit table. Clearing condition A value of 1 is written to ULEC |

CAUTION

ADCGnULER is updated when the A/D converted value is written to ADCGnDRj.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.31 ADCGnOWER — Overwrite Error Register

ADCGnOWER is an 8-bit read-only register that indicates an overwrite error.

Access: This register can be read only in 8-bit units.

Address: <ADCGn_base> + 3E0_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------------|-----|---|-------------|---|---|---|---|---|--|
| | OWE | — | OWECAP[5:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | |

Table 30.37 ADCGnOWER Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 7 | OWE | Overwrite Error 0: No error 1: An error is present. Setting condition The A/D converted value is written to ADCGnDRj when WFLG = 1 Clearing condition A value of 1 is written to OWEC |
| 6 | Reserved | When read, the value after reset is read. |
| 5 to 0 | OWECAP[5:0] | Overwrite Error Capture The virtual channel at the time when an overwrite error occurred is captured. Capturing condition The A/D converted value is written to ADCGnDRj when OWE = 0 and WFLG = 1. Clearing condition A value of 1 is written to OWEC |

CAUTION

ADCGnOWER is updated when the A/D converted value is written to ADCGnDRj.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.32 ADCGnPER — Parity Error Register

ADCGnPER is an 8-bit read-only register that indicates a parity error.

Access: This register can be read only in 8-bit units.

Address: <ADCGn_base> + 3E4_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------------|----|---|------------|---|---|---|---|---|--|
| | PE | — | PECAP[5:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | |

Table 30.38 ADCGnPER Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 7 | PE | Parity Error 0: No error 1: An error is present. Setting condition A parity error is detected. Clearing condition A value of 1 is written to PEC. |
| 6 | Reserved | When read, the value after reset is read. |
| 5 to 0 | PECAP[5:0] | Parity Error Capture The virtual channel at the time when a parity error occurred is captured. Capturing condition A parity error is detected when PE = 0. Clearing condition A value of 1 is written to PEC. |

CAUTION

ADCGnPER is updated when ADCGnDRj or ADCGnDIRj is read.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.33 ADCGnIDER — ID Error Register

ADCGnIDER is an 8-bit read-only register that indicates an ID error.

Access: This register can be read only in 8-bit units.

Address: <ADCGn_base> + 3E8_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------------|-----|---|-------------|---|---|---|---|---|--|
| | IDE | — | IDECAP[5:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | |

Table 30.39 ADCGnIDER Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 7 | IDE | ID Error 0: No error 1: An error is present. Setting condition The physical channel specified in ADCGnVCRj does not match the physical channel actually converted. Clearing condition A value of 1 is written to IDEC. |
| 6 | Reserved | When read, the value after reset is read. |
| 5 to 0 | IDECAP[5:0] | ID Error Capture The virtual channel at the time when an ID error occurred is captured. Capturing condition The physical channel specified in ADCGnVCRj does not match the physical channel actually converted when IDE = 0. Clearing condition A value of 1 is written to IDEC. |

CAUTION

ADCGnIDER is updated when the A/D converted value is written to ADCGnDRj.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.34 ADCGnSGSTCRx — Scan Group x Start Control Register

ADCGnSGSTCRx is an 8-bit write-only register that controls the start of scan group x. The register bits are always read as 0.

Access: This register can be written only in 8-bit units.

Address: ADCGnSGSTCR0: <ADCGn_base> + 480_H
 ADCGnSGSTCR1: <ADCGn_base> + 500_H
 ADCGnSGSTCR2: <ADCGn_base> + 580_H
 ADCGnSGSTCR3: <ADCGn_base> + 600_H
 ADCGnSGSTCR4: <ADCGn_base> + 680_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|------|
| | — | — | — | — | — | — | — | SGST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.40 ADCGnSGSTCRx Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SGST | Starts A/D Conversion of Scan Group x 0: No function (Writing 0 is ignored.) 1: Start A/D conversion while A/D conversion start scan group status (ADCGnSGSRx.SGACT) is 0. |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.35 ADCGnADTSTCRy — A/D Timer y Start Control Register

ADCGnADTSTCRy is an 8-bit write-only register that controls the start of A/D timer y. The register bits are always read as 0.

Access: This register can be written only in 8-bit units.

Address: ADCGnADTSTCR3: <ADCGn_base> + 608_H
ADCGnADTSTCR4: <ADCGn_base> + 688_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|-------|
| | — | — | — | — | — | — | — | ADTST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.41 ADCGnADTSTCRy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ADTST | Starts A/D timer y. 0: No function (Writing 0 is ignored.) 1: A/D timer is started. Start A/D timer while A/D timer status (ADCGnSGSRx.ADTACT) is 0. |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.36 ADCGnADTENDCRy — A/D Timer y End Control Register

ADCGnADTENDCRy is an 8-bit write-only register that controls the end of the A/D timer y. The register bits are always read as 0.

Access: This register can be written only in 8-bit units.

Address: ADCGnADTENDCR3: <ADCGn_base> + 60C_H
ADCGnADTENDCR4: <ADCGn_base> + 68C_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|--------|
| | — | — | — | — | — | — | — | ADTEND |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 30.42 ADCGnADTENDCRy Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ADTEND | Ends A/D timer. 0: No function (Writing 0 is ignored.) 1: End A/D timer. |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.37 ADCGnSGCRx — Scan Group x Control Register

ADCGnSGCRx is an 8-bit readable/writable register that controls scan group x.

Access: This register can be read/written in 8-bit units.

Address: ADCGnSGCR0: <ADCGn_base> + 490_H
 ADCGnSGCR1: <ADCGn_base> + 510_H
 ADCGnSGCR2: <ADCGn_base> + 590_H
 ADCGnSGCR3: <ADCGn_base> + 610_H
 ADCGnSGCR4: <ADCGn_base> + 690_H

Value after reset: 00_H

- When x = 0 to 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|----------|--------|------|---|---|---|----------|
| | — | ADSTARTE | SCANMD | ADIE | — | — | — | TRGMD[0] |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R | R | R | R/W |

Table 30.43 ADCGnSGCRx Register Contents (x = 0 to 2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | ADSTARTE | Scan Group Synchronization Start Enable 0: ADSTART is disabled. 1: ADSTART is enabled. |
| 5 | SCANMD | Scan Mode 0: Multicycle scan mode 1: Continuous scan mode In multicycle scan mode, scans are repeated as many times as specified in ADCGnSGMCYCRx. In continuous scan mode, scans are repeated with no limit of times. |
| 4 | ADIE | Scan End Interrupt Enable 0: INTADCGnIx is not output at the end of scan for SGx. 1: INTADCGnIx is output at the end of scan for SGx. ADIE of ADCGnSGCRx is independent of ADIE of ADCGnVCRj. For details, see Section 30.4.5.1, Scan End Interrupt Request . |
| 3 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | TRGMD[0] | Enables or disables trigger input to San Group x |

- x = 0

| TRGMD[0] | Hardware trigger | Software trigger |
|----------|----------------------|----------------------|
| | Trigger input to SGx | Trigger input to SGx |
| 0 | Disabled | Enabled |
| 1 | Enabled | Enabled |

- x = 1, 2

| TRGMD[0] | Hardware trigger | | Software trigger | |
|----------|----------------------|---------------------|----------------------|---------------------|
| | Trigger input to SGx | Hold trigger to SGx | Trigger input to SGx | Hold trigger to SGx |
| 0 | Disabled | Disabled | Enabled | Disabled |
| 1 | Enabled | Enabled | Enabled | Enabled |

- When $x = 3, 4$

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|-----------|----------|--------|------|---|---|------------|-----|
| | ADTSTARTE | ADSTARTE | SCANMD | ADIE | — | — | TRGMD[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W |

Table 30.44 ADCGnSGCRx Register Contents ($x = 3, 4$)

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 7 | ADTSTARTE | A/D Timer Synchronization Start Enable 0: ADTSTART is disabled. 1: ADTSTART is enabled. |
| 6 | ADSTARTE | Scan Group Synchronization Start Enable 0: ADSTART is disabled. 1: ADSTART is enabled. |
| 5 | SCANMD | Scan Mode 0: Multicycle scan mode 1: Continuous scan mode In multicycle scan mode, scans are repeated as many times as specified in ADCGnSGMCCRx. In continuous scan mode, scans are repeated unlimited of times. |
| 4 | ADIE | Scan End Interrupt Enable 0: INTADCGnIx is not output at the end of scan for SGx. 1: INTADCGnIx is output at the end of scan for SGx. ADIE of ADCGnSGCRx is independent of ADIE of ADCGnVCRj. For details, see Section 30.4.5.1, Scan End Interrupt Request . |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | TRGMD[1:0] | Enables or disables trigger input to Scan Group x. |

| TRGMD [1:0] | Hardware trigger | | | Software trigger | | |
|-------------|----------------------|--------------------------|---------------------|----------------------|--------------------------|---------------------|
| | Trigger input to SGx | Trigger input to timer x | Hold trigger to SGx | Trigger input to SGx | Trigger input to timer x | Hold trigger to SGx |
| 0 | Disabled | Disabled | Disabled | Enabled | Disabled | Disabled |
| 1 | Enabled | Disabled | Enabled | Enabled | Disabled | Enabled |
| 2 | Disabled | Disabled | Disabled | Enabled | Enabled | Disabled |
| 3 | Disabled | Enabled | Disabled | Enabled | Enabled | Disabled |

CAUTIONS

- To prevent malfunctions, make settings for SCANMD and ADIE of ADCGnSGCRx after making or confirming the following settings.
 - HLDTE of the T&H group A is 0 when scan group x is selected in SGS[1:0] of the T&H group A.
 - HLDTE of the T&H group B is 0 when scan group x is selected in SGS[1:0] of the T&H group B.
 - ADTSTARTE of scan group x is 0 and TRGMD of scan group x is 0_H.
 - SGACT of scan group x is 0 (before the scan group is started).
- If a trigger of lower-priority scan group is input to a scan group for which continuous scan mode is set (SCANMD = 1_H), the trigger is not accepted. Therefore, it is assumed that continuous scan mode is specified for scan group 0.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.38 ADCGnSGVCSPx — Scan Group x Start Virtual Channel Pointer

ADCGnSGVCSPx is an 8-bit readable/writable register that specifies the start pointer of a virtual channel.

Access: This register can be read/written in 8-bit units.

Address: ADCGnSGVCSP0: <ADCGn_base> + 494_H
 ADCGnSGVCSP1: <ADCGn_base> + 514_H
 ADCGnSGVCSP2: <ADCGn_base> + 594_H
 ADCGnSGVCSP3: <ADCGn_base> + 614_H
 ADCGnSGVCSP4: <ADCGn_base> + 694_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|-----------|-----|-----|-----|-----|-----|
| | — | — | VCSP[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.45 ADCGnSGVCSPx Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | VCSP[5:0] | Start Virtual Channel Pointer These bits select the virtual channel from which the scan is to be started. When SGx is started, processing for the virtual channels from ADCGnSGVCSPx to ADCGnSGVCEPx is executed. |

CAUTIONS

- ADCGnSGVCSPx must be equal to or smaller than ADCGnSGVCEPx.
- To prevent malfunctions, make ADCGnSGVCSPx settings after making or confirming the following settings.
 - HLDTE of the T&H group A is 0 when scan group x is selected in SGS[1:0] of the T&H group A.
 - HLDTE of the T&H group B is 0 when scan group x is selected in SGS[1:0] of the T&H group B.
 - ADSTARTE of scan group x is 0 and TRGMD of scan group x is 0_H.
 - SGACT of scan group x is 0 (before the scan group is started).
- Do not set a value greater than the number of virtual channels provided.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.39 ADCGnSGVCEPx — Scan Group x End Virtual Channel Pointer

ADCGnSGVCEPx is an 8-bit readable/writable register that specifies the end pointer of a virtual channel.

Access: This register can be read/written in 8-bit units.

Address: ADCGnSGVCEP0: <ADCGn_base> + 498_H
 ADCGnSGVCEP1: <ADCGn_base> + 518_H
 ADCGnSGVCEP2: <ADCGn_base> + 598_H
 ADCGnSGVCEP3: <ADCGn_base> + 618_H
 ADCGnSGVCEP4: <ADCGn_base> + 698_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|-----------|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | VCEP[5:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.46 ADCGnSGVCEPx Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 to 0 | VCEP[5:0] | End Virtual Channel Pointer These bits select the virtual channel at which the scan is to end. When SGx is started, processing for the virtual channels from ADCGnSGVCSPx to ADCGnSGVCEPx is executed. |

CAUTION

To prevent malfunctions, make ADCGnSGVCEPx settings after making or confirming the following settings.

- (1) HLDTE of the T&H group A is 0 when scan group x is selected in SGS[1:0] of the T&H group A.
- (2) HLDTE of the T&H group B is 0 when scan group x is selected in SGS[1:0] of the T&H group B.
- (3) ADSTARTE of scan group x is 0 and TRGMD of scan group x is 0_H.
- (4) SGACTION of scan group x is 0 (before the scan group is started).

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.40 ADCGnSGMICYCRx — Scan Group x Multicycle Register

ADCGnSGMICYCRx is an 8-bit readable/writable register that specifies the number of scan modes.

Access: This register can be read/written in 8-bit units.

Address: ADCGnSGMICYCR0: <ADCGn_base> + 49C_H
 ADCGnSGMICYCR1: <ADCGn_base> + 51C_H
 ADCGnSGMICYCR2: <ADCGn_base> + 59C_H
 ADCGnSGMICYCR3: <ADCGn_base> + 61C_H
 ADCGnSGMICYCR4: <ADCGn_base> + 69C_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MCYC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.47 ADCGnSGMICYCRx Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 0 | MCYC[7:0] | Multicycle These bits specify the number of scan times in multicycle scan mode. Number of scan times = MCYC[7:0] + 1 When SGx is started, scans are repeated for virtual channels from ADCGnSGVCSPx to ADCGnSGVCEPx as many times as specified in ADCGnSGMICYCRx. |

CAUTION

To prevent malfunctions, make ADCGnSGMICYCRx settings after making or confirming the following settings.

- (1) HLDTE of the T&H group A is 0 when scan group x is selected in SGS[1:0] of the T&H group A.
- (2) HLDTE of the T&H group B is 0 when scan group x is selected in SGS[1:0] of the T&H group B.
- (3) ADSTARTE of scan group x is 0 and TRGMD of scan group x is 0_H.
- (4) SGACT of scan group x is 0 (before the scan group is started).

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.41 ADCGnSGSRx — Scan Group x Status Register

ADCGnSGSRx is an 8-bit read-only register that indicates the status of scan group x.

Access: This register can be read only in 8-bit units.

Address: ADCGnSGSR0: <ADCGn_base> + 4A4_H
 ADCGnSGSR1: <ADCGn_base> + 524_H
 ADCGnSGSR2: <ADCGn_base> + 5A4_H
 ADCGnSGSR3: <ADCGn_base> + 624_H
 ADCGnSGSR4: <ADCGn_base> + 6A4_H

Value after reset: 00_H

- x = 0 to 2

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|-------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | SGACT | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 30.48 ADCGnSGSRx Register Contents (x = 0 to 2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. |
| 1 | SGACT | Scan Group Status 0: Scan Group is in idle state. 1: Scan Group is during A/D conversion. |
| 0 | Reserved | When read, the value after reset is read. |

- x = 3, 4

| | | | | | | | | |
|-------------------|---|---|---|---|---|--------|-------|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | ADTACT | SGACT | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 30.49 ADCGnSGSRx Register Contents (x = 3, 4)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 3 | Reserved | When read, the value after reset is read. |
| 2 | ADTACT | A/D Timer Status 0: A/D timer x is in idle state. 1: A/D timer x is running |
| 1 | SGACT | Scan Group Status 0: Scan Group is in idle state. 1: Scan Group is during A/D conversion. |
| 0 | Reserved | When read, the value after reset is read. |

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.42 ADCGnADTIPRy — A/D Timer Initial Phase Register y

ADCGnADTIPRy is a 32-bit readable/writable register that sets the initial phase of A/D timer y.

Access: This register can be read/written in 32-bit units.

Address: ADCGnADTIPR3: <ADCGn_base> + 628_H
ADCGnADTIPR4: <ADCGn_base> + 6A8_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | | | ADTIP[20:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ADTIP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.50 ADCGnADTIPRy Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 0 | ADTIP[20:0] | A/D Timer Initial Phase These bits set the initial phase of A/D timer y. (1) After A/D timer y is started, ADCGnADTIPRy is loaded to A/D timer y and the timer counts down. (2) After A/D timer y becomes 0, A/D timer y trigger is output for one cycle, ADCGnADTPRRy is loaded to A/D timer y, and the timer counts down again. After that, (2) is repeated. For details, see Section 30.4.2.3, Starting A/D Timer by Using a Hardware Trigger . |

CAUTION

To prevent malfunctions, make ADCGnADTIPRy settings when ADTACT of scan group y is 0 (before A/D timer is started), ADTSTARTE of scan group y is 0, and TRGMD[1:0] of scan group y is not 3_H.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.43 ADCGnADTPRRy — A/D Timer Period Register y

ADCGnADTPRRy is a 32-bit readable/writable register that sets the cycle of A/D timer y.

Access: This register can be read/written in 32-bit units.

Address: ADCGnADTPRR3: <ADCGn_base> + 62C_H
ADCGnADTPRR4: <ADCGn_base> + 6AC_H

Value after reset: 001F FFFF_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------------|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | ADTPR[20:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ADTPR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 30.51 ADCGnADTPRRy Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 0 | ADTPR[20:0] | <p>A/D Timer Cycle</p> <p>These bits set the cycle of A/D timer y.</p> <p>(1) After A/D timer y is started, ADCGnADTIPRy is loaded to A/D timer y and the timer counts down.</p> <p>(2) After A/D timer y becomes 0, A/D timer y trigger is output for one cycle, ADCGnADTPRRy is loaded to A/D timer y, and the timer counts down again.</p> <p>After that, (2) is repeated.</p> <p>For details, see Section 30.4.2.3, Starting A/D Timer by Using a Hardware Trigger.</p> |

CAUTION

To prevent malfunctions, make ADCGnADTPRRy settings when ADTACT of scan group y is 0 (before A/D timer is started), ADTSTARTE of scan group y is 0, and TRGMD[1:0] of scan group y is not 3_H.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.3.44 ADCGnULLMSRx — Scan Group x Upper-limit/Lower-limit Table Select Register

ADCGnULLMSRx is an 8-bit readable/writable register that controls scan group x.

Access: This register can be read/written in 8-bit units.

Address: ADCGnULLMSR0: <ADCGn_base> + 4B0_H
 ADCGnULLMSR1: <ADCGn_base> + 530_H
 ADCGnULLMSR2: <ADCGn_base> + 5B0_H
 ADCGnULLMSR3: <ADCGn_base> + 630_H
 ADCGnULLMSR4: <ADCGn_base> + 6B0_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|----------|-----|
| | — | — | — | — | — | — | ULS[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R/W | R/W |

Table 30.52 ADCGnULLMSRx Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | ULS[1:0] | Upper Limit/Lower Limit Table Select 0 _H : Neither upper limit nor lower limit is checked. 1 _H : Upper limit and lower limit are checked in ADCGnULLMTBR0. 2 _H : Upper limit and lower limit are checked in ADCGnULLMTBR1. 3 _H : Upper limit and lower limit are checked in ADCGnULLMTBR2. Upper limit and lower limit are checked by using the upper limit/lower limit table selected by ULS[1:0] when storing the A/D converted value in ADCGnDRj. |

CAUTION

To prevent malfunctions, make ADCGnULLMSRx settings after making or confirming the following settings.

- (1) HLDTE of the T&H group A is 0 when scan group x is selected in SGS[1:0] of the T&H group A.
- (2) HLDTE of the T&H group B is 0 when scan group x is selected in SGS[1:0] of the T&H group B.
- (3) ADSTARTE of scan group x is 0 and TRGMD of scan group x is 0_H.
- (4) SGACT of scan group x is 0 (before the scan group is started).

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.4 Function

30.4.1 A/D Conversion Function

A/D Conversion is performed in scan group units. All scan groups have A/D conversion trigger signals. When a trigger signal (SG_x_TRG) is input, the signals on the virtual channels assigned to the scan group are A/D converted in ascending order. An A/D complete interrupt (INTADCGnIx) is generated on completion of A/D conversion for each virtual channel or for all virtual channels assigned to a scan group.

In multi-scan mode, A/D conversion is repeated the specified number of times in response to the input of a trigger signal. In continuous scan mode, A/D conversion is repeated until it is stopped.

The A/D conversion data format can be selected between signed fixed-point format and signed integer format. For the bit sequence of the data formats, see **Section 30.3.5, ADCGnDRj — Data Register j**.

30.4.1.1 Example of Multicycle Scan Mode Operation

The following figure shows an example of operation when converting four virtual channels by 2-cycle scan for scan group 0 in multicycle scan mode by using normal A/D conversion mode (CNVCLS[2:0] = 0_H).

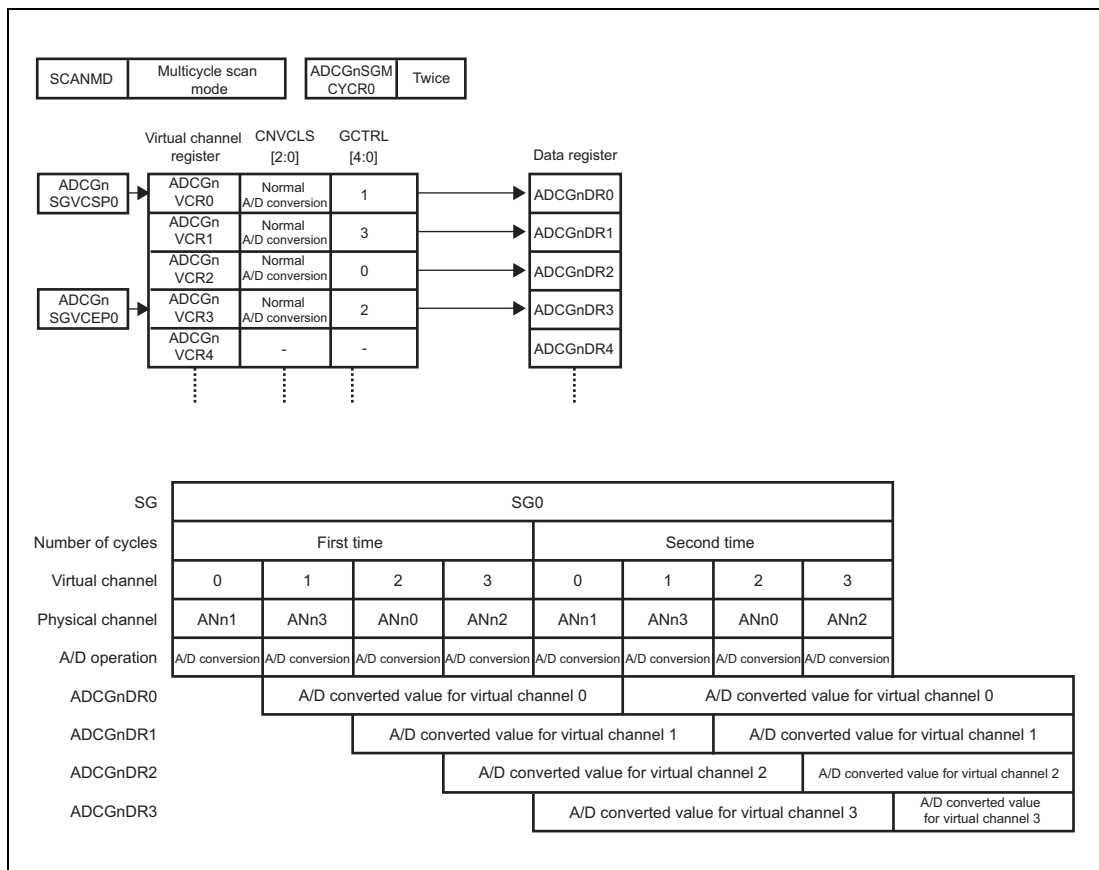


Figure 30.4 Example of Operation in Multicycle Scan Mode

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.4.1.2 Example of Continuous Scan Mode Operation

Figure 30.5 shows an example of operation when converting four virtual channels for scan group 0 in continuous scan mode by using normal A/D conversion mode (CNVCLS[2:0] = 0_H).

In this mode, if a trigger of a lower-priority scan group is input to a scan group for which continuous scan mode is set (SCANMD = 1_H), the trigger is not accepted. Therefore, it is assumed that continuous scan mode is set for scan group 0.

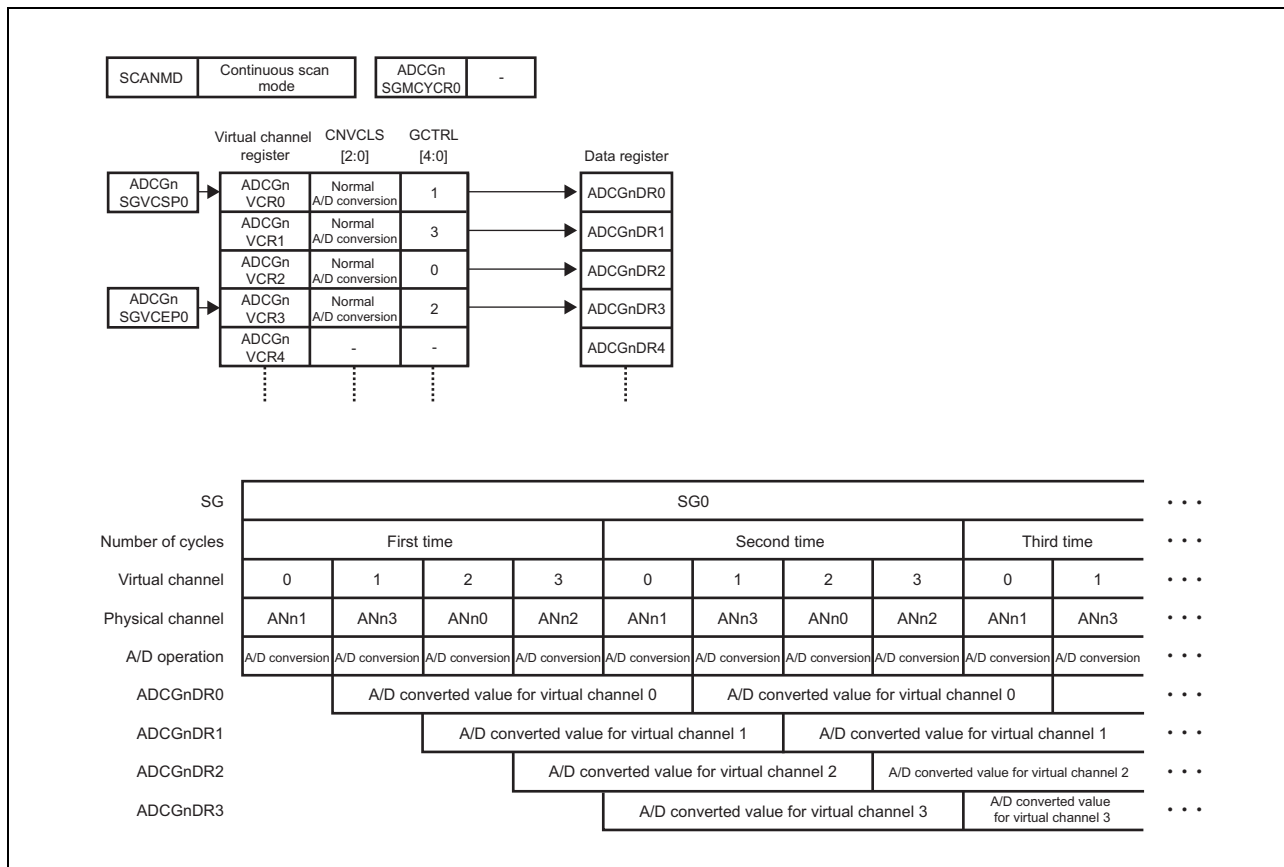


Figure 30.5 Example of Operation in Continuous Scan Mode

30.4.1.3 Simultaneous Track and Hold Operation (THC Control)

The following figures show examples of simultaneous track and hold operation with T&H control, A/D conversion of the held value (CNVCLS[2:0] = 1_H), self-diagnosis (CNVCLS[2:0] = 3_H), and normal A/D conversion in addition mode (CNVCLS[2:0] = 4_H) by THC.

If a hold trigger is input from T&H group A or B, on-going conversion of the virtual channel is suspended regardless of the priority of the scan group and the hold operation is started. When the hold operation is completed, the virtual channel operation is executed in order of priority of the scan groups. The virtual channel suspended by the hold trigger is restarted from the beginning.

The following figures (**Figure 30.6, Figure 30.7, Figure 30.8, and Figure 30.9**) show the operations when a hold trigger is generated during A/D conversion.

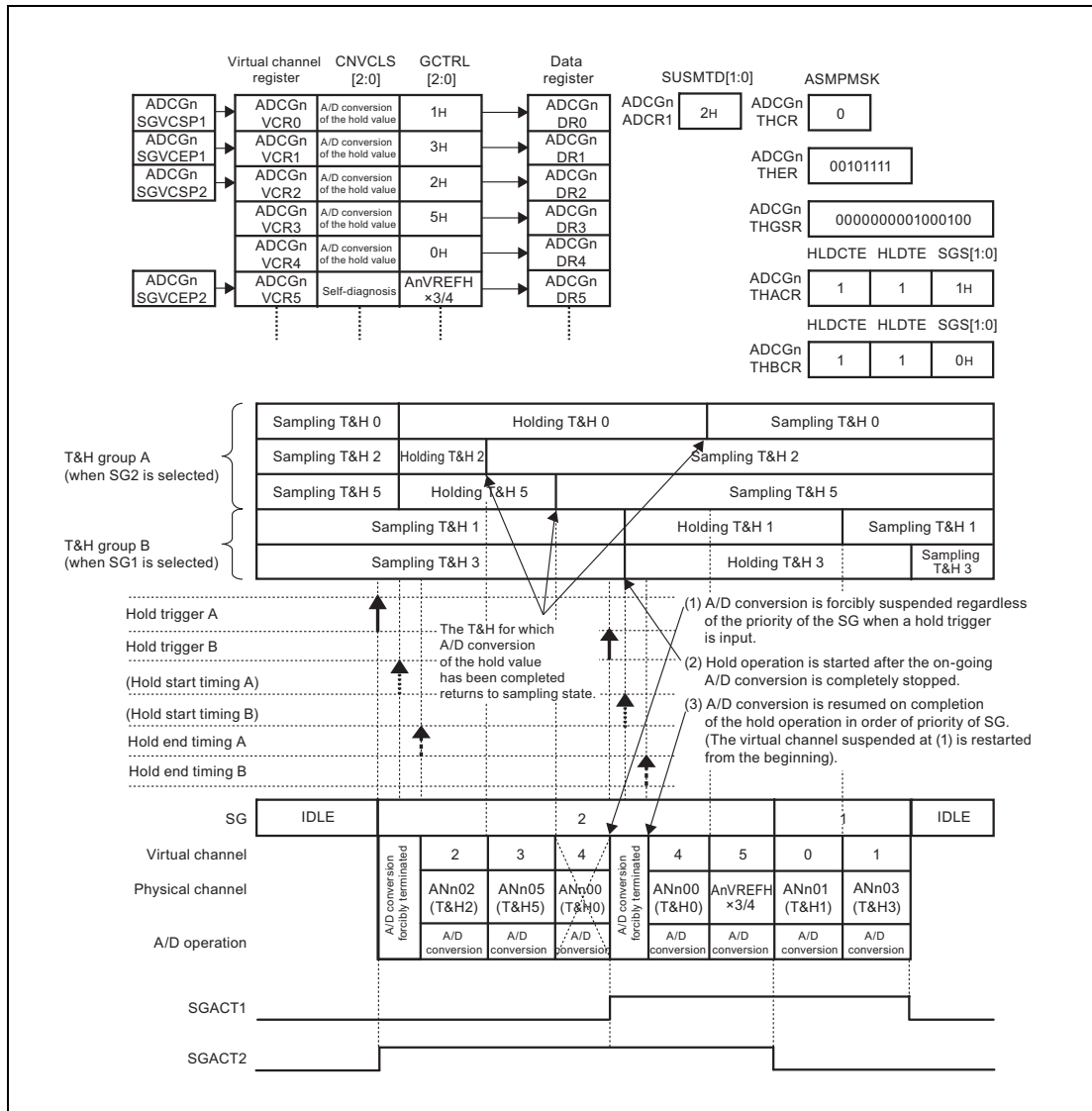


Figure 30.6 Simultaneous Track and Hold Operation (THC Control/Case 1)

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units.**

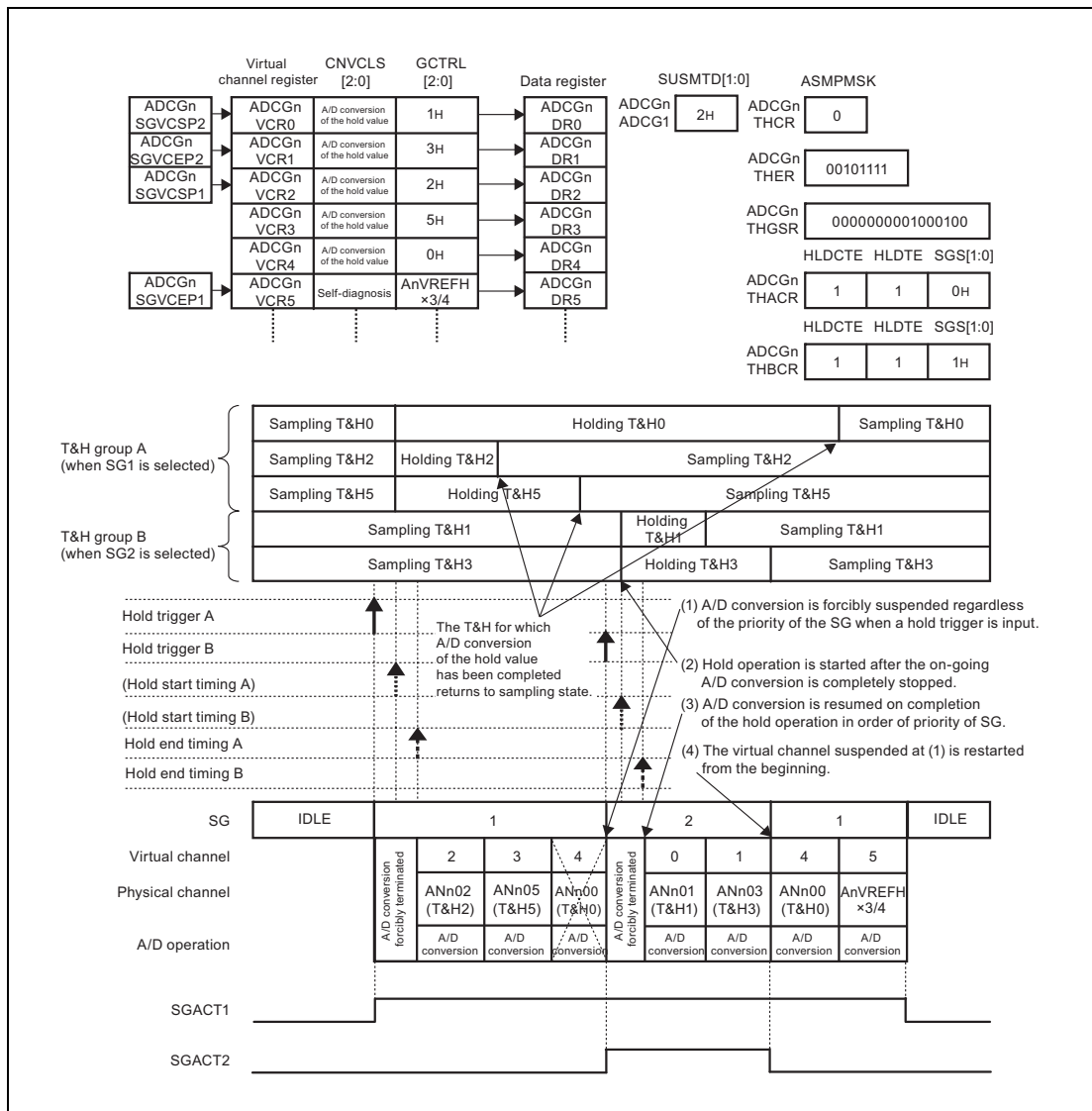


Figure 30.7 Simultaneous Track and Hold Operation (THC Control/Case 2)

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

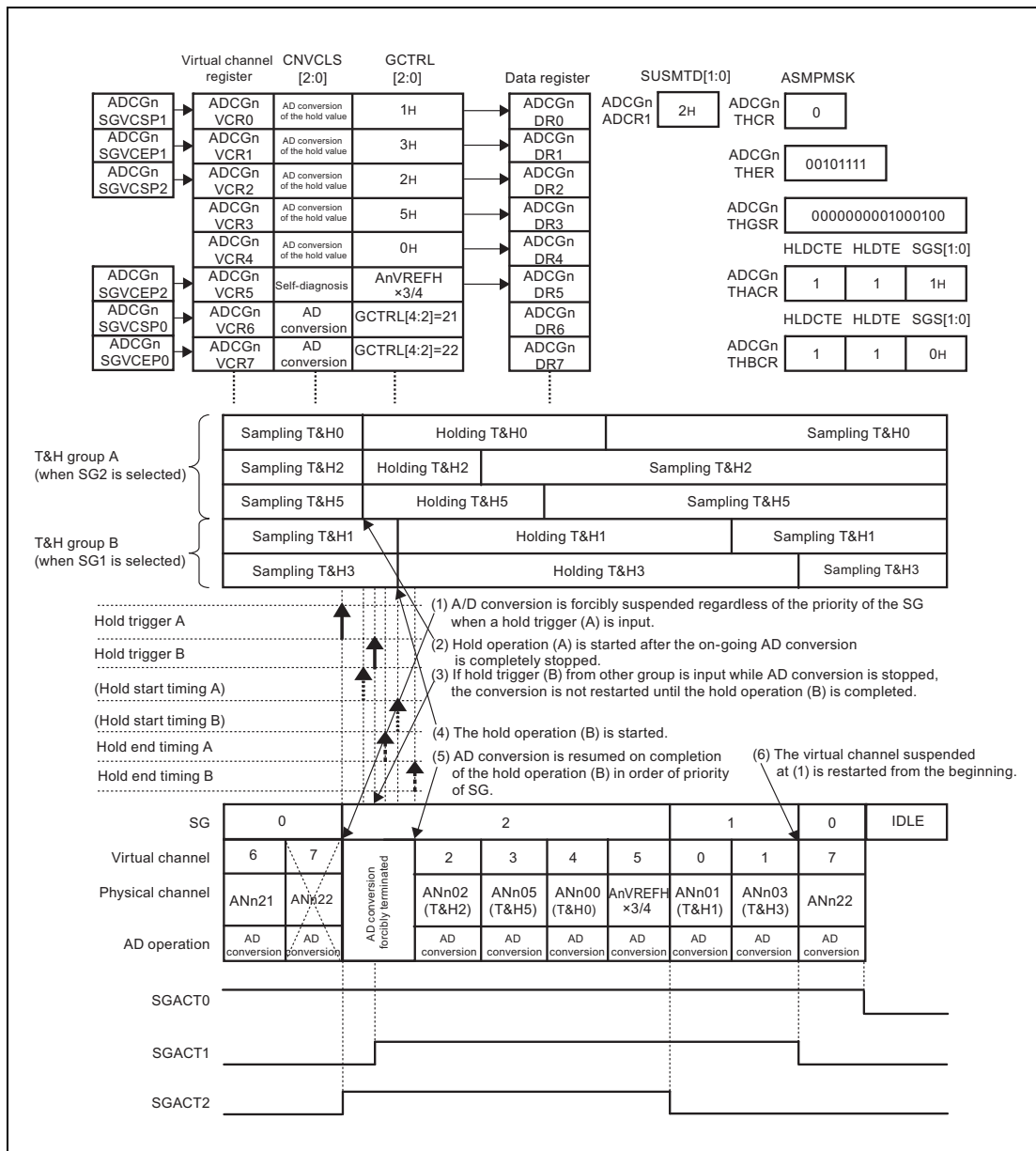


Figure 30.8 Simultaneous Track and Hold Operation (THC Control/Case 3)

NOTE

For the number of units and indices, see Section 30.1.1, Number of Units.

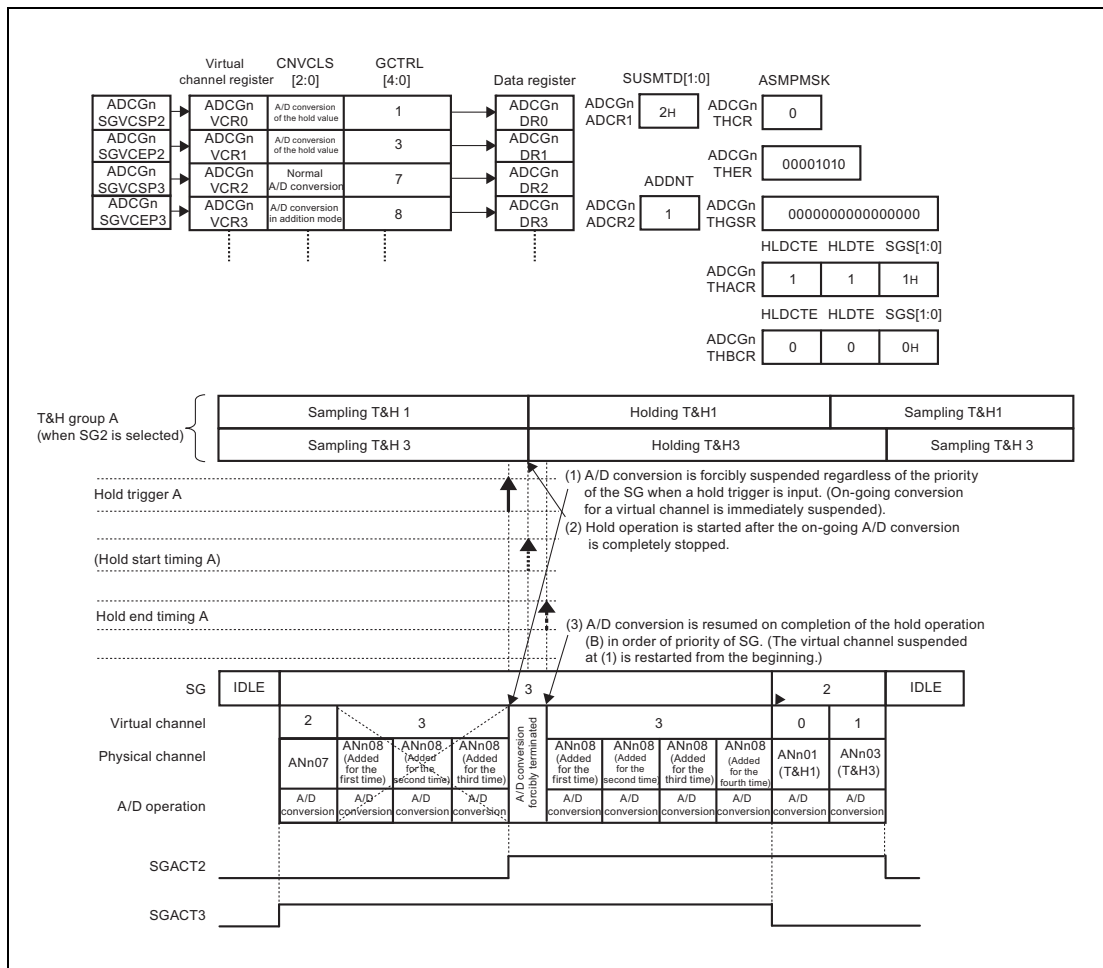


Figure 30.9 Simultaneous Track and Hold Operation (THC Control/Case 4)

NOTE

For the number of units and indices, see Section 30.1.1, Number of Units.

30.4.1.4 Example of Normal A/D Conversion Operation in Addition Mode

Figure 30.10 shows an example of operation when converting four virtual channels for scan group 0 by using normal A/D conversion (CNVCLS[2:0] = 4_H) in addition mode.

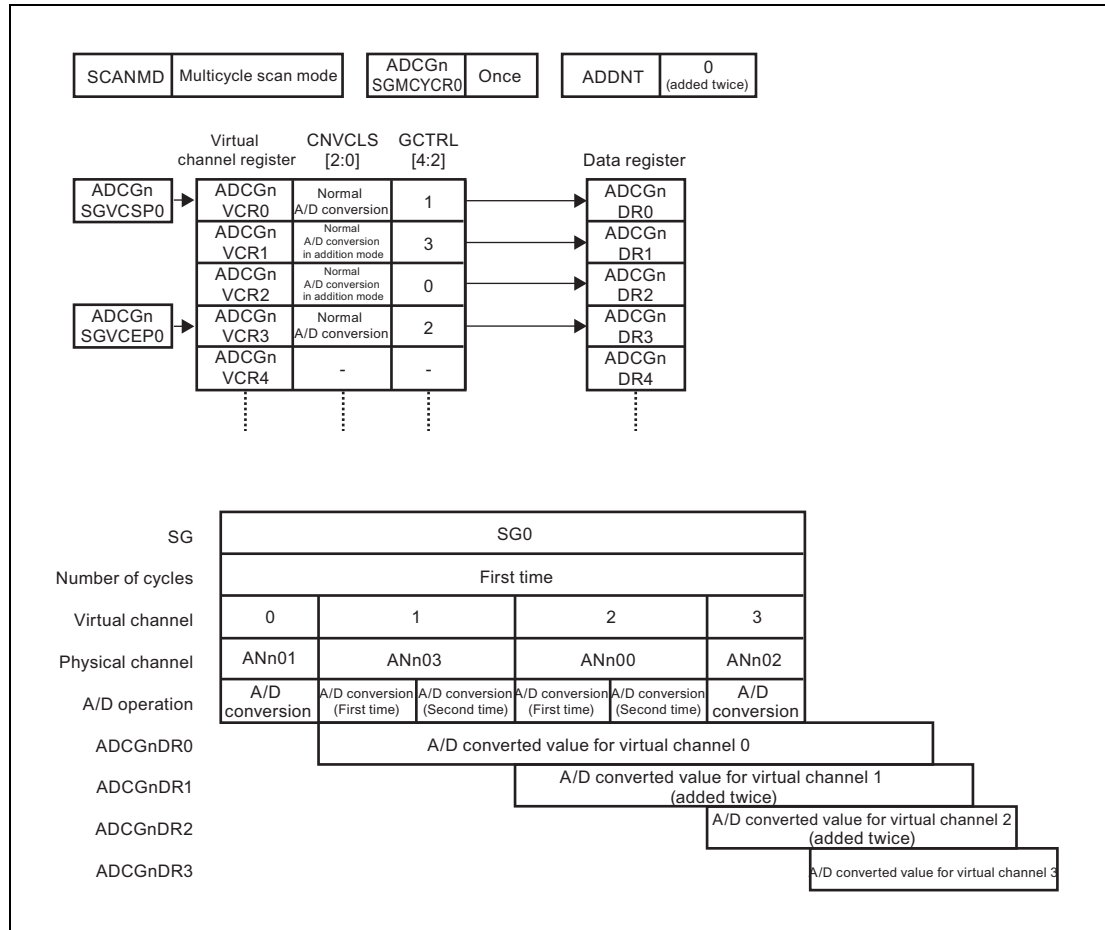


Figure 30.10 Example of Normal A/D Conversion Operation in Addition Mode

30.4.1.5 Example of Using an External Analog Multiplexer (Port Output)

Figure 30.11 shows an example of port output using an external analog multiplexer.

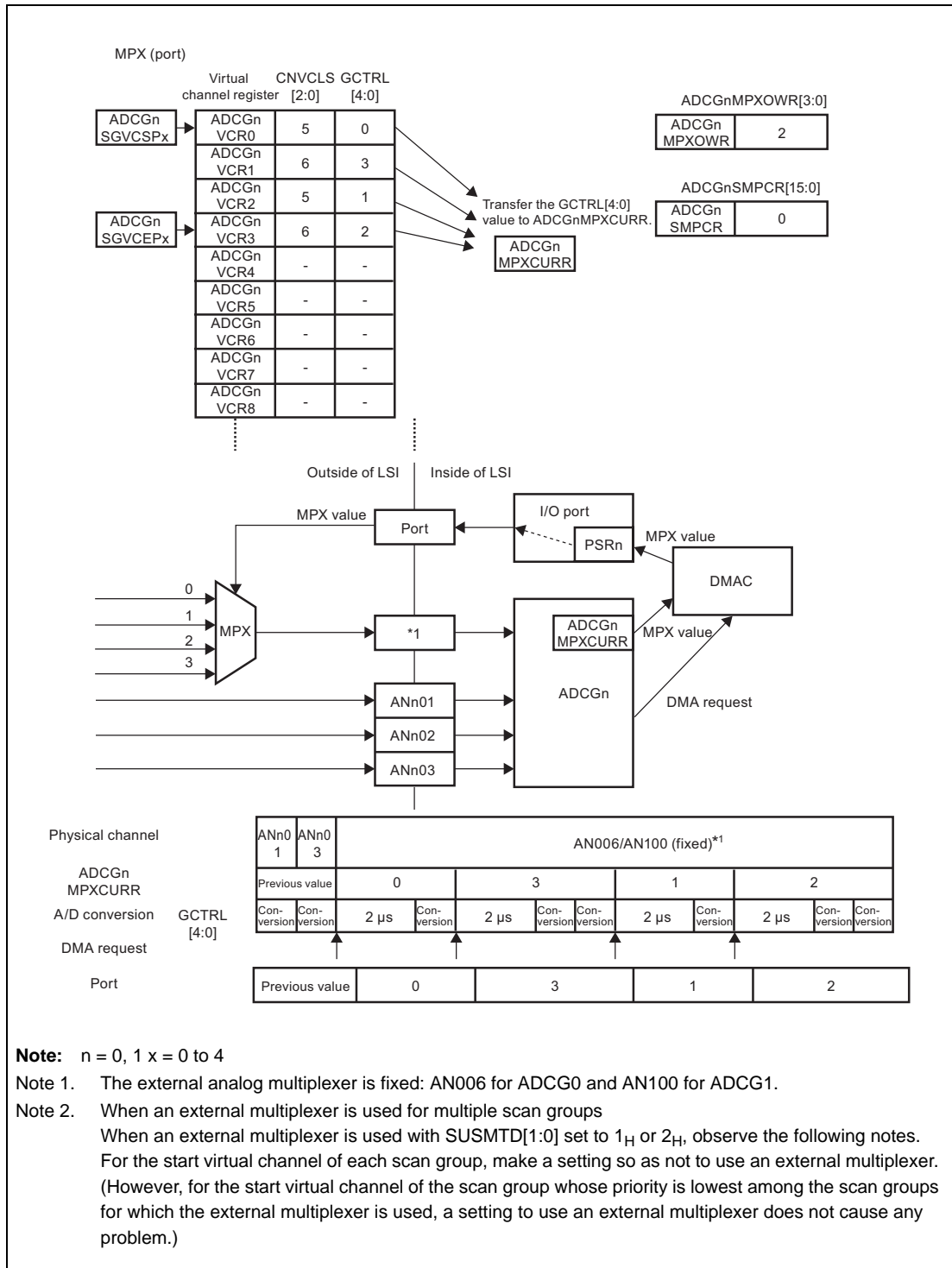


Figure 30.11 Example of Using an External Analog Multiplexer (Port Output)

30.4.2 Trigger Function

The following triggers are selectable as an A/D conversion start trigger of each scan group.

- Hardware (SG_x_TRG)*¹
- A/D timer
- Software (by setting the ADCGnSGSTCR_x)

The following triggers are selectable for activating A/D timer.

- Hardware (SG_x_TRG)*¹
- Software (by setting the ADCGnADTSTCR_y)

Triggering of a scan group which is the same as the scan group being A/D converted is invalid.

Triggering of a scan group which has a lower-priority than the scan group being A/D converted is put on hold; the conversion is started after the conversion of higher-priority scan groups are completed.

For triggering of a scan group which has a higher-priority than the scan group being A/D converted, see **Section 30.4.3, Suspend Function**.

Note 1. For details on SG_x_TRG, see **Section 29.3, Peripheral Interconnection — 2 (PIC2B)**.

30.4.2.1 Starting a Scan Group by Using a Hardware Trigger

Scan group *x* can be started by using an input from hardware trigger SG_x_TRG. To start scan group *x* by an input from hardware trigger SG_x_TRG, set TRGMD in ADCGnSGCR_x to 1_H. When the selected hardware trigger SG_x_TRG is input in this state, SGACT is set to 1. The timing since SGACT is set to 1 until scan group *x* is started is the same as the timing when SGST is set to 1 by a software trigger.

NOTES

1. For the number of units and indices, see **Section 30.1.1, Number of Units**.
2. When a hardware trigger SG_x_TRG is generated during A/D conversion (ADCGnSGSR_x.SGACT=1) for same scan group *x*, the hardware trigger SG_x_TRG is ignored.

30.4.2.2 Starting a Scan Group by Using an A/D Timer Trigger

Scan group 3 or 4 can be started by using a trigger from A/D timer 3 or 4. To start scan group 3 or 4 by using a trigger from A/D timer 3 or 4, set TRGMD in ADCGnSGCR3 or ADCGnSGCR4 to 2_H. Furthermore, set ADTST of scan group 3 or 4 to 1 to start A/D timer 3 or 4.

When a timer trigger is input in this state, SGACT is set to 1. The timing since SGACT is set to 1 until scan group 3 or 4 is started is the same as the timing when SGST is set to 1 by a software trigger.

30.4.2.3 Starting A/D Timer by Using a Hardware Trigger

A/D timer 3 or 4 can be started by using a hardware trigger SG3_TRG or SG4_TRG input. To start A/D timer 3 or 4 by using a hardware trigger SG3_TRG or SG4_TRG input, set TRGMMD in ADCGnSGCR3 or ADCGnSGCR4 to 3_H. When the selected external trigger is input in this state, A/D timer 3 or 4 starts. Furthermore, a trigger from A/D timer 3 or 4 starts scan group 3 or 4.

Figure 30.12 shows an example of A/D timer operation.

The A/D timer counts in synchronization with the CLK_ADC clock.

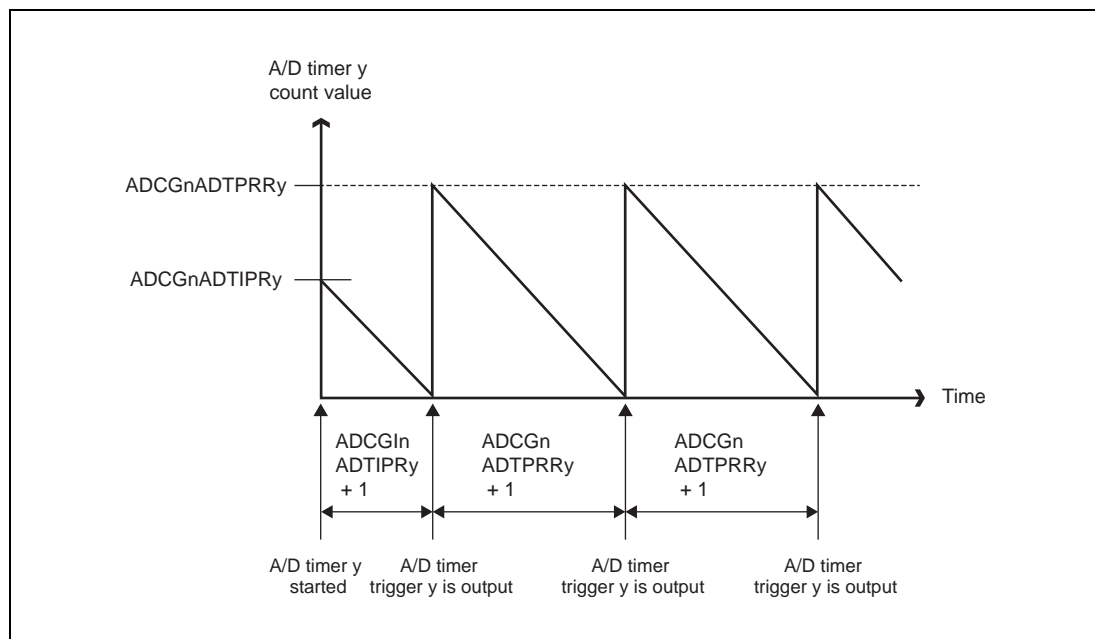


Figure 30.12 Example of A/D Timer Operation

- (1) A/D timer y is loaded with ADCGnADTIPRy at the start of counting.
- (2) If the A/D timer y is 0, an A/D timer trigger is output for one cycle and the A/D timer y is loaded with ADCGnADTPRRy.

If the A/D timer y is not 0, the timer counts down. After that, (2) is repeated.

CAUTION

If 0 is set to the A/D timer initial phase register y (ADCGnADTIPRy), an A/D timer trigger is output simultaneously with the start of the A/D timer.

If 0 is set to the A/D timer period register y (ADCGnADTPRRy), an A/D timer trigger is output at each clock cycle.

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.4.3 Suspend Function

In suspend function, when a scan group request with higher priority is issued during lower-priority scan group processing, A/D conversion of the lower priority level is suspended and A/D conversion with higher priority level is performed. There are three types in the suspend operation.

30.4.3.1 Example of Synchronous Suspend and Resume Operation

When the A/D conversion trigger is generated for a scan group with higher priority than one on which A/D conversion is in progress, A/D conversion for the scan group with higher priority does not proceed until A/D conversion of the virtual channel for which conversion was in progress is completed. A/D conversion for the suspended scan group is resumed from the virtual channel after the one that was being converted on completion of A/D conversion for the scan group with higher priority.

Figure 30.13 shows an example of synchronous suspend and resume operation when a higher-priority SG interrupts a lower-priority SG.

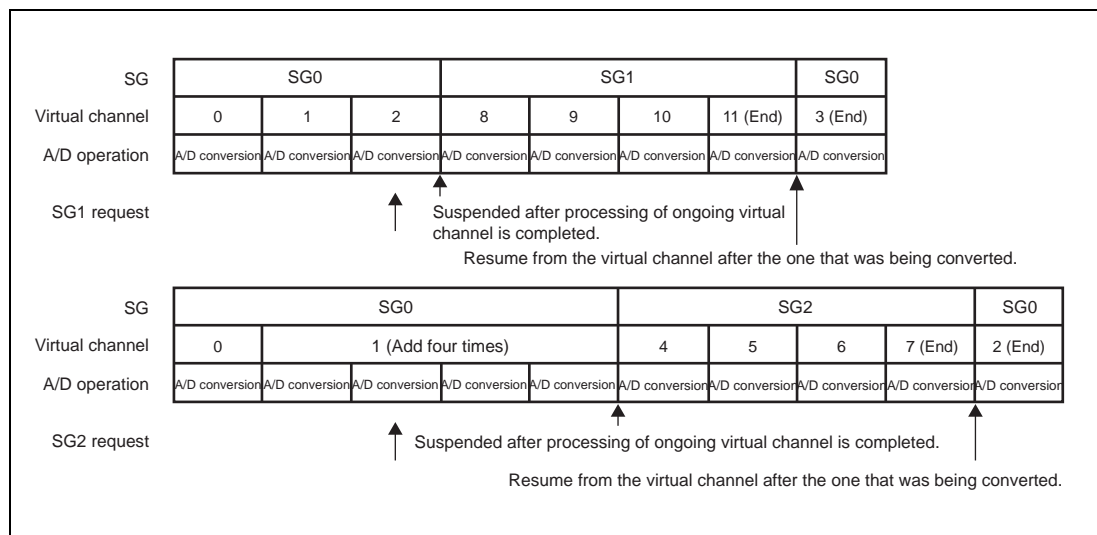


Figure 30.13 Example of Synchronous Suspend and Resume Operation

CAUTION

Priority of scan groups:
 Low High
 SG0 < SG1 < SG2 < SG3 < SG4

30.4.3.2 Example of Asynchronous Suspend and Resume Operation

When the A/D conversion trigger is generated for a scan group with higher priority than one on which A/D conversion is in progress, A/D conversion of the virtual channel for which conversion was in progress is suspended immediately and A/D conversion for the scan group with higher priority proceeds. A/D conversion for the suspended scan group is resumed on completion of A/D conversion for the scan group with higher priority.

Figure 30.14 shows an example of asynchronous suspend and resume operation when a higher priority SG interrupts a lower-priority SG.

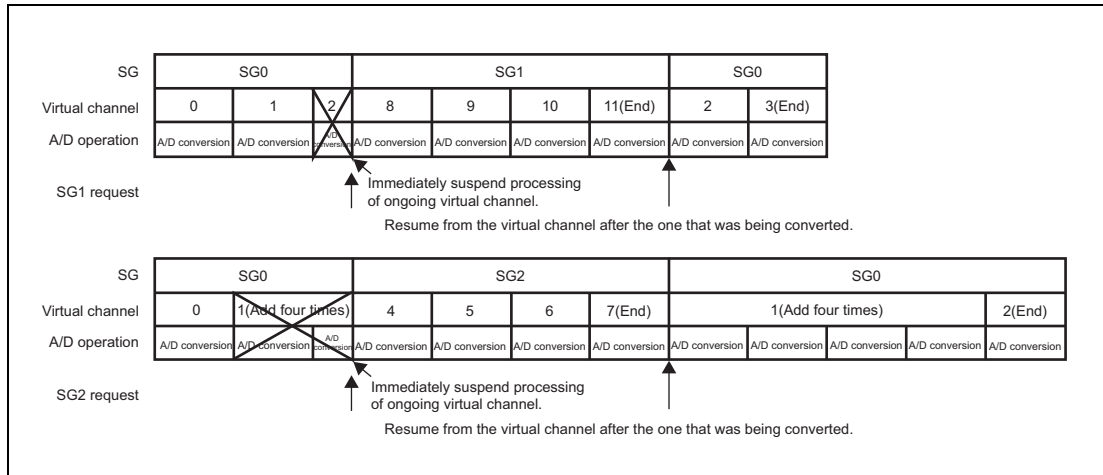


Figure 30.14 Example of Asynchronous Suspend and Resume Operation

CAUTION

Priority of scan groups:

Low High
 SG0 < SG1 < SG2 < SG3 < SG4

30.4.3.3 Mixed Synchronous and Asynchronous Suspend Operations

If scan group 0 is interrupted by a higher priority scan group, an asynchronous suspend operation is performed. If a scan group other than scan group 0 is interrupted by a higher priority scan group, a synchronous suspend operation is performed.

An example of mixed synchronous and asynchronous suspend operations is shown below.

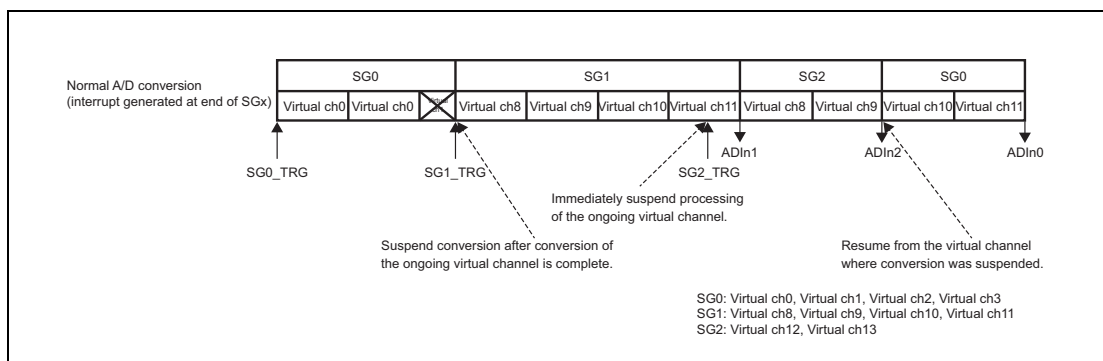


Figure 30.15 Example of Mixed Synchronous and Asynchronous Suspend Operations

30.4.4 Self-Diagnostic Function

The ADCG is equipped with the following self-diagnostic functions.

- Pin level self-diagnostic function
- A/D conversion circuit self-diagnostic function
- Wiring-break detection function

30.4.4.1 Pin Level Self-Diagnostic Function

Figure 30.16 is a flowchart of pin-level self-diagnosis.

The pin-level self-diagnosis flowchart below illustrates an example where pin-level self-diagnosis proceeds after assigning all ADCG0 pins to virtual channels and switching the applied voltage to even and odd numbered physical channel groups. This flowchart assumes that pin-level self-diagnosis proceeds before the start of A/D conversion.

[Features]

1. Physical channels for diagnosis can be arbitrarily selected.
2. The voltage level of pin-level self-diagnosis (AnVSS, AnVCC, or $1/2 \times \text{AnVCC}$) can be selected for each even- or odd-numbered physical channel.
3. Pin-level self-diagnosis function is available by performing A/D conversion for SG0 to SG4.

[Settings]

1. Make initial settings according to **Figure 30.24, Initial Settings**.
2. In the virtual channel register ADCGnVCRj (j = 0 to 23), set normal A/D conversion ($\text{CNVCLS}[2:0] = 0_H$) and set the desired physical channel for pin-level self-diagnosis in $\text{GCTRL}[5:0]$.
3. In pin-level self-diagnosis control register ADCGnTDCR, set diagnosis voltage ($\text{TDLV}[1:0]$) and set pin-level self-diagnosis ($\text{TDE} = 1$).
4. Configure other necessary settings for A/D conversion according to **Figure 30.24, Initial Settings**.
5. Assert the trigger for SG0 to SG4 to start A/D conversion.

NOTE

The results of conversion of diagnosis voltage at self-diagnosis include the total errors at selfdiagnosis (TESH0SN). For the values of TESH0SN, see **Section 37.14, A/D Converter Characteristics**.

CAUTION

The accuracy of conversion at a pin-level self-diagnosis is affected by the current injected. For details, see Section 37.14, A/D Converter Characteristics.

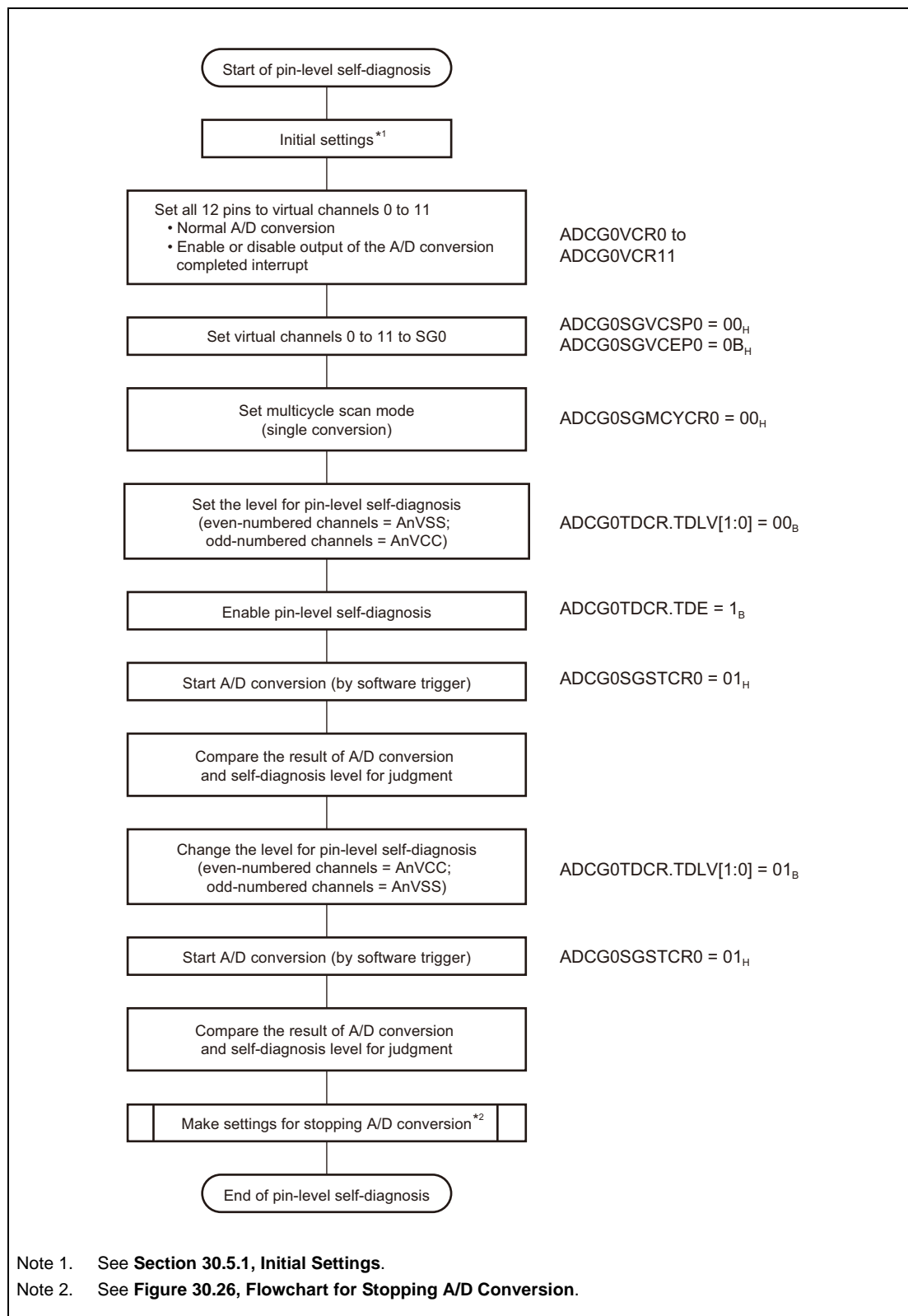


Figure 30.16 Flowchart of Pin-Level Self-Diagnosis (Example of ADCG0 144-pin Package)

30.4.4.2 A/D Conversion Circuit Self-Diagnostic Function

The A/D conversion circuit self-diagnostic function is used to verify that A/D conversion operates correctly.

The voltage value setting is made in GCTRL[4:0] when CNVCLS[2:0] = 3_H, and can be converted for AnVREFH×1, AnVREFH×3/4, AnVREFH×1/2, AnVREFH×1/4, and AnVSS.

Features and settings of the A/D conversion circuit self-diagnostic function of the ADCG are described below.

[Features]

1. AnVREFH×1, AnVREFH×3/4, AnVREFH×1/2, AnVREFH×1/4, or AnVSS can be selected as a self-diagnosis voltage level.
2. Performing A/D conversion for SG0 to SG4 makes the A/D conversion circuit self-diagnostic function available.

[Settings]

1. Make initial settings according to **Figure 30.24, Initial Settings**.
2. Set self-diagnosis mode (CNVCLS[2:0] = 3_H) and any self-diagnosis voltage level with GCTRL[5:0] in the virtual channel register ADCGnVCRj (j = 0 to 23).
3. Make other settings required for A/D conversion according to **Figure 30.24, Initial Settings**.
4. Assert triggers of SG0 to SG4 to start A/D conversion.

NOTE

The results of conversion of diagnosis voltage at self-diagnosis include the total errors at selfdiagnosis (TESH0SN). For the values of TESH0SN, see **Section 37.14, A/D Converter Characteristics**.

CAUTION

Do not use the temperature sensor when A/D conversion circuit self-diagnostic function is used. Set TSN0CR.TSNEN = 0.

30.4.4.3 Wiring-Break Detection

Wiring-break detection is a facility for detecting wiring breaks in ANnm. If a wiring break exists the result of A/D conversion converges approximately to AnVSS or AnVCC due to pull-down or pull-up resistor. Therefore, the existence of wiring-breaks can be determined from the conversion result.

ADCGn supports two wiring-break detection modes.

- Mode 1: A wiring-break detection for each physical channel can be performed at the same time as normal conversion (wiring-break detection by pull-down resistor).
- Mode 2: A wiring-break detection for each virtual channel can be performed by using the function dedicated to wiring-break (wiring-break detection by pull-down or pull-up resistor).

(1) Wiring-break detection mode 1 (ADCGnODCR.ODE = 1)

In wiring-break detection mode 1, the desired width of pull-down pulse (1 to 17 states) can be set and a pull-down resistor is connected for conversion by successive approximation (A pull-up resistor cannot be used in wiring-break detection mode 1). Enabling or disabling of wiring-break detection is specified for each analog pin.

[Settings]

1. Make initial settings according to **Figure 30.24, Initial Settings**.
2. In virtual channel register ADCGnVCRj (j = 0 to 23), set normal A/D conversion (CNVCLS[2:0] = 0_H), set a pull-up/-down resistor off (ADCGnVCRj.PDE = 0_H, and ADCGnVCRj.PUE = 0_H), and set the desired channel in GCTRL[5:0].
3. Set wiring-break detection mode 1 (ADCGnODCR.ODE = 1), the physical channel for wiring-break detection (ADOPDIGnm = 1), and the desired width of pull-down pulse in ADCGnODCR.ODPW[5:0].
4. Assert the trigger signal for SG0 to SG4 to start A/D conversion.
5. Repeat A/D conversion several times to check that the result voltage of A/D conversion has not been converged approximately to AnVSS.

(2) Wiring-break detection mode 2 (ADCGnODCR.ODE = 0)

In wiring-break detection mode 2, a pull-down or pull-up resistor is connected for sampling of the sample-and-hold circuits when the width of pulse is fixed (18 states). Enabling or disabling of wiring-break detection can be set for each virtual channel.

[Settings]

1. Make initial settings according to **Figure 30.24, Initial Settings**.
2. In virtual channel register ADCGnVCRj (j = 0 to 23) for the virtual channel where wiring-break is to be detected, set normal A/D conversion mode (CNVCLS[2:0] = 0_H), set a pull-down or pull-up resistor on (ADCGnVCRj.PDE = 1_H, or ADCGnVCRj.PUE = 1_H) and set the desired channel in GCTRL[5:0].
3. Set wiring-break detection mode 2 (ADCGnODCR.ODE = 0) and the physical channel for wiring-break detection (ADOPDIGnm = 1).
4. Assert the trigger signal for SG0 to SG4 to start A/D conversion.

5. Repeat A/D conversion several times to check that the result voltage of A/D conversion with the pull-down resistor has not been converged approximately to AnVSS and the voltage of the result of A/D conversion with the pull-up resistor has not been converged approximately to AnVCC.

30.4.4.4 Diagnosis of Wiring-Break Detection

Diagnosis of wiring-break detection is a self-diagnosis facility to check that wiring-break detection is working properly.

Start by performing A/D conversion in pin-level self-diagnosis mode, and then calculate the voltage which will be used as a baseline for comparison.

Next, pseudo-disconnect ANnm and repeat A/D conversion from several to a few dozen times using wire-break detection mode 2. If the result of A/D conversion in this disconnected state converged approximately to AnVSS or AnVCC, you can determine that wiring-break detection is working properly.

[Features]

1. Physical channels for diagnosis of wiring-break detection can be arbitrarily selected.

[Settings]

1. Make initial settings according to **Figure 30.24, Initial Settings**.
2. In the virtual channel register ADCGnVCRj, set normal A/D conversion mode (CNVCLS[2:0] = 0_H), and the desired physical channel for wiring-break detection in GCTRL[5:0].
3. In the pin-level self-diagnosis control register ADCGnTDCR, set pin-level self-diagnosis (TDE = 1_H) and the desired level for pin-level self-diagnosis in TDLV[1:0].
4. Assert the trigger signal for SG0 to SG4 to start A/D conversion.
5. After A/D conversion, enable wiring-break detection diagnosis (ODDE = 1_H) in ADCGnODCR (ADCGn automatically turns off pin-level self-diagnosis).
6. In ADCGnVCRj, set a pull-down resistor on (PDE = 1_H) or pull-up resistor on (PUE = 1_H), set wire-break detection mode 2 (ADCGnODCR.ODE = 0), and set the desired physical channel for wiring-break detection (ADOPDIGnm = 1).
7. Assert the trigger signal for SG0 to SG4 to start A/D conversion.
8. Repeat A/D conversion from several to a few dozen times in wiring-break detection diagnosis mode to check that the result of A/D conversion has converged approximately to AnVSS for the pull-down resistor and AnVCC for the pull-up resistor.

30.4.4.5 Pin-Level Self-Diagnosis of T&H Paths

A facility for pin-level self-diagnosis is used for self-diagnostic testing of T&H paths.

This checks that the results of pin-level self-diagnosis for both the voltage held by the T&H circuit and the voltage when the T&H circuit is not used match the diagnostic voltages. If this test is run for all values (0_H , 1_H , 2_H , and 3_H) of the pin-level self-diagnosis level specification bits (TDLV) of the pin-level self-diagnosis control register (ADCGnTDCR) and the results match the respective diagnostic voltages, the T&H paths can be considered normal.

The following describes the procedures for configuration and evaluation when **(1)** pin-level self-diagnosis proceeds with the voltages held by the T&H circuits being used and when **(2)** pin-level self-diagnosis proceeds with the voltages held by the T&H circuits not being used.

NOTE

The settings in (1) and (2) below assume that the even- and odd-numbered physical channel groups are connected as follows.

- Even-numbered physical channel group = ANn00, ANn02, ANn04, ANn06, ...
- Odd-numbered physical channel group = ANn01, ANn03, ANn05, ANn07, ...

(1) Pin-level self-diagnosis of T&H paths (the voltage held by the T&H circuits used)

[Settings]

1. Make initial settings as shown in **Figure 30.18**.
2. Write 1 to the sampling start bit (SMPST) of the T&H sampling start control register (ADCGnTHSMPSTCR) to start sampling by the T&H circuits.
3. Write 1 to the scan group start bit (SGST) of the scan group 2 start control register (ADCGnSGSTCR2) to start A/D conversion of the signals in scan group 2 by software trigger.
4. Write 01_H to the hold start bit (HLDST) of the hold start control register (ADCGnTHAHLSTCR) to cause the T&H circuits of group A to hold the input voltages.*1
5. Write 1 to the scan group start bit (SGST) of the scan group 1 start control register (ADCGnSGSTCR1) to start conversion of the signals in scan group 1.
6. Wait for the first A/D conversion completed interrupt from ADCGnVCR0 for scan group 2 and write 1_H to the pin-level self-diagnosis level specification bit (TDLV) of the pin-level self-diagnosis control register (ADCGnTDCR) while A/D conversion is in progress to change the self-diagnosis level.
7. After the A/D conversion completed interrupt from ADCGnVCR5 for scan group 1, check the results of conversion by scan group 1.

If the results in the respective data registers (ADCGnDRj) of the even- and odd-numbered physical channel groups all match the diagnostic voltage when tested with the pin-level self-diagnosis level specification bits (TDLV) set to all values from 0_H to 3_H , the results of the pin-level self-diagnosis when the T&H circuits are used are considered normal. Also, the results of conversion by scan group 2 are not used but discarded.

- Note 1. Wait for at least 32 cycles of CLK_ADC after the start of sampling by T&H circuits in step (1) before setting the hold start control register in step (3) in Figure 30.16, Example of Timing Chart for Self-Diagnosis of T&H Paths (the voltage held by the T&H circuits used) (TDLV = $0_H \rightarrow 1_H$)

NOTE

The results of conversion of diagnosis voltage at self-diagnosis include the total errors at selfdiagnosis (TESH0SN). For the values of TESH0SN, see **Section 37.14, A/D Converter Characteristics**.

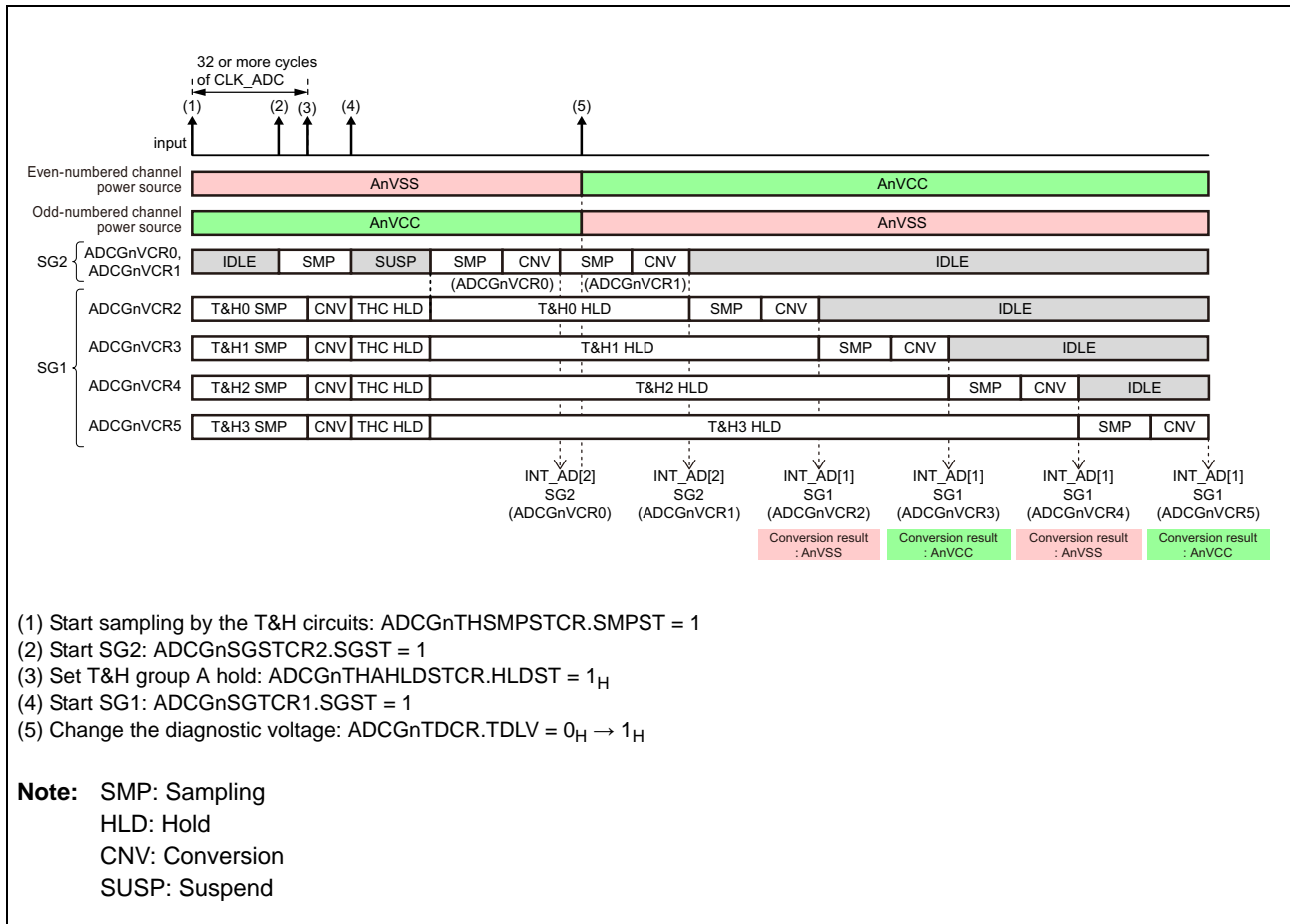


Figure 30.17 Example of Timing Chart for Self-Diagnosis of T&H Paths (the voltage held by the T&H circuits used) (TDLV = 0_H → 1_H)

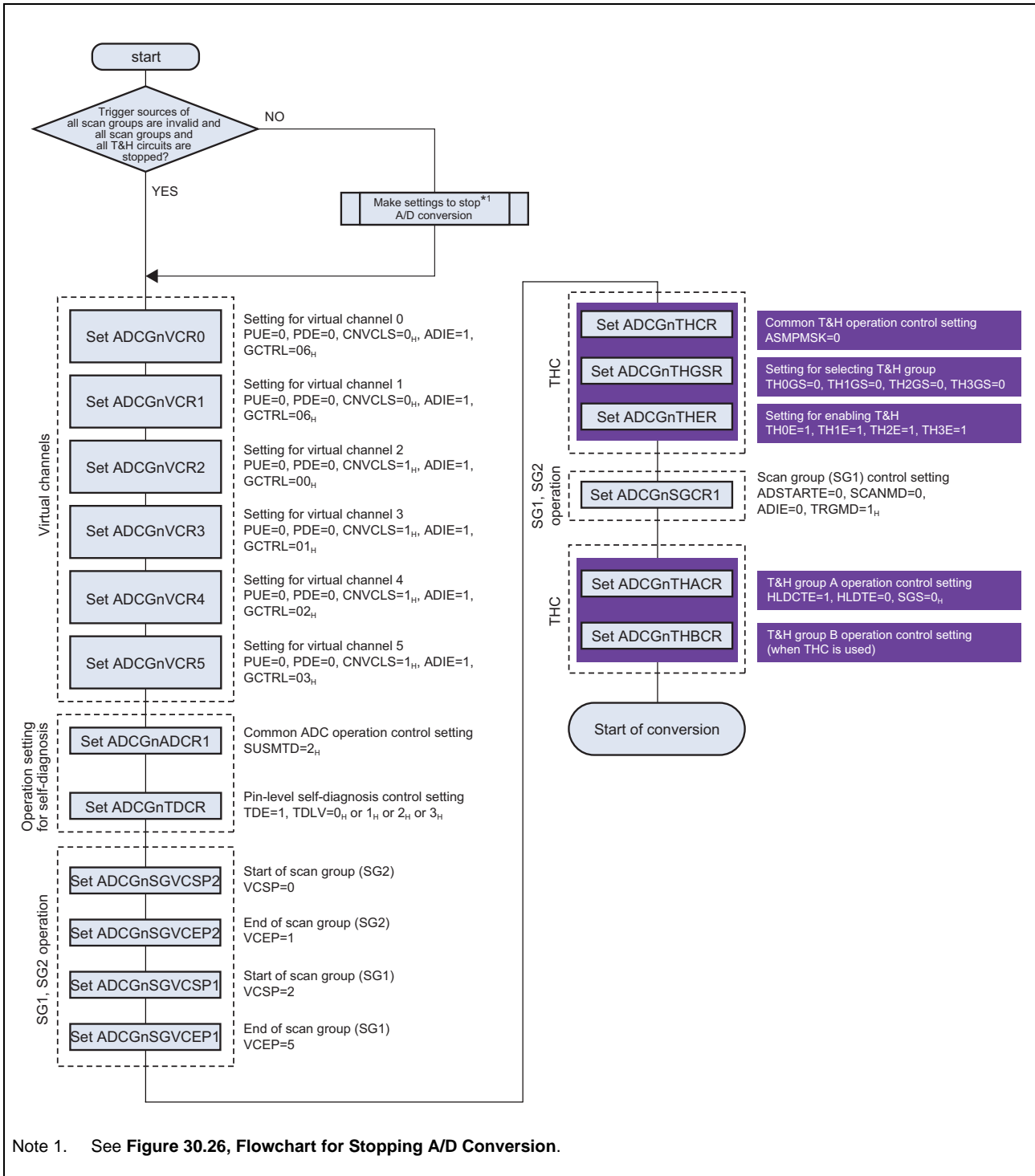


Figure 30.18 Initial Settings for Self-Diagnosis of T&H Paths (the voltage held by the T&H circuits used)

(2) Pin-level self-diagnosis of T&H paths (the voltage held by the T&H circuits not used)**[Settings]**

1. Make initial settings as shown in **Figure 30.20**.
2. Write 1 to the sampling start bit (SMPST) of the T&H sampling start control register (ADCGnTHSMPSTCR) to start sampling by the T&H circuits.
3. Write 1 to the scan group start bit (SGST) of the scan group 2 start control register (ADCGnSGSTCR2) as the software trigger for A/D conversion of the signals in scan group 2.
4. Write 01_H to the hold start bit (HLDST) of the hold start control register (ADCGnTHAHLSTCR) to cause the T&H circuits of group A to hold the input voltages.*¹
5. Write 1 to the scan group start bit (SGST) of the scan group 1 start control register (ADCGnSGSTCR1) to start conversion of the signals in scan group 1.
6. Wait for the first A/D conversion completed interrupt from ADCGnVCR0 for scan group 2 and write 1_H to the pin-level self-diagnosis level specification bit (TDLV) of the pin-level self-diagnosis control register (ADCGnTDCR) while A/D conversion is in progress to change the self-diagnosis level.
7. After the A/D conversion completed interrupt from ADCGnVCR5 for scan group 1, check the results of conversion by scan group 1.
If the respective data registers (ADCGnDRj) hold the results of A/D conversion of the self-diagnosis level voltages by the even- and odd-numbered physical channel groups when tested with the pin-level self-diagnosis level specification bits (TDLV) set to all values from 0_H to 3_H, the results of the pin-level self-diagnosis when the T&H circuits are not used are considered normal. Also, the results of conversion by scan group 2 are not used but discarded.

Note 1. Wait for at least 32 cycles of CLK_ADC after the start of sampling by T&H circuits in step (1) before setting the hold start control register in step (3) in **Figure 30.17, Example of Timing Chart for Self-Diagnosis of T&H Paths (the voltage held by the T&H circuits used) (TDLV = 0_H → 1_H)**.

NOTE

The results of conversion of diagnosis voltage at self-diagnosis include the total errors at selfdiagnosis (TESH0SN). For the values of TESH0SN, see **Section 37.14, A/D Converter Characteristics**.

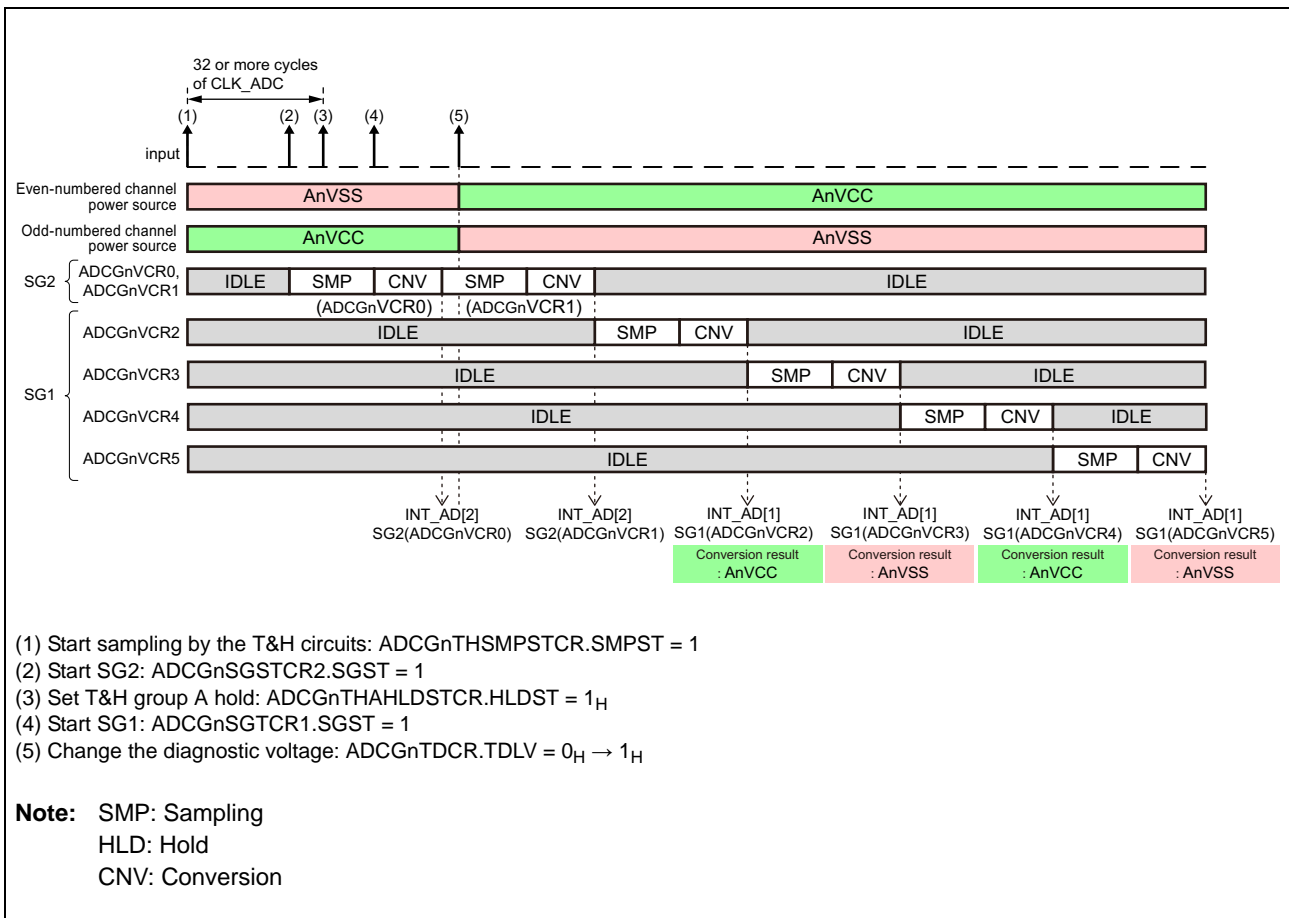


Figure 30.19 Example of Timing Chart for Self-Diagnosis of T&H Paths (the voltage held by the T&H circuits not used) (TDLV = 0_H → 1_H)

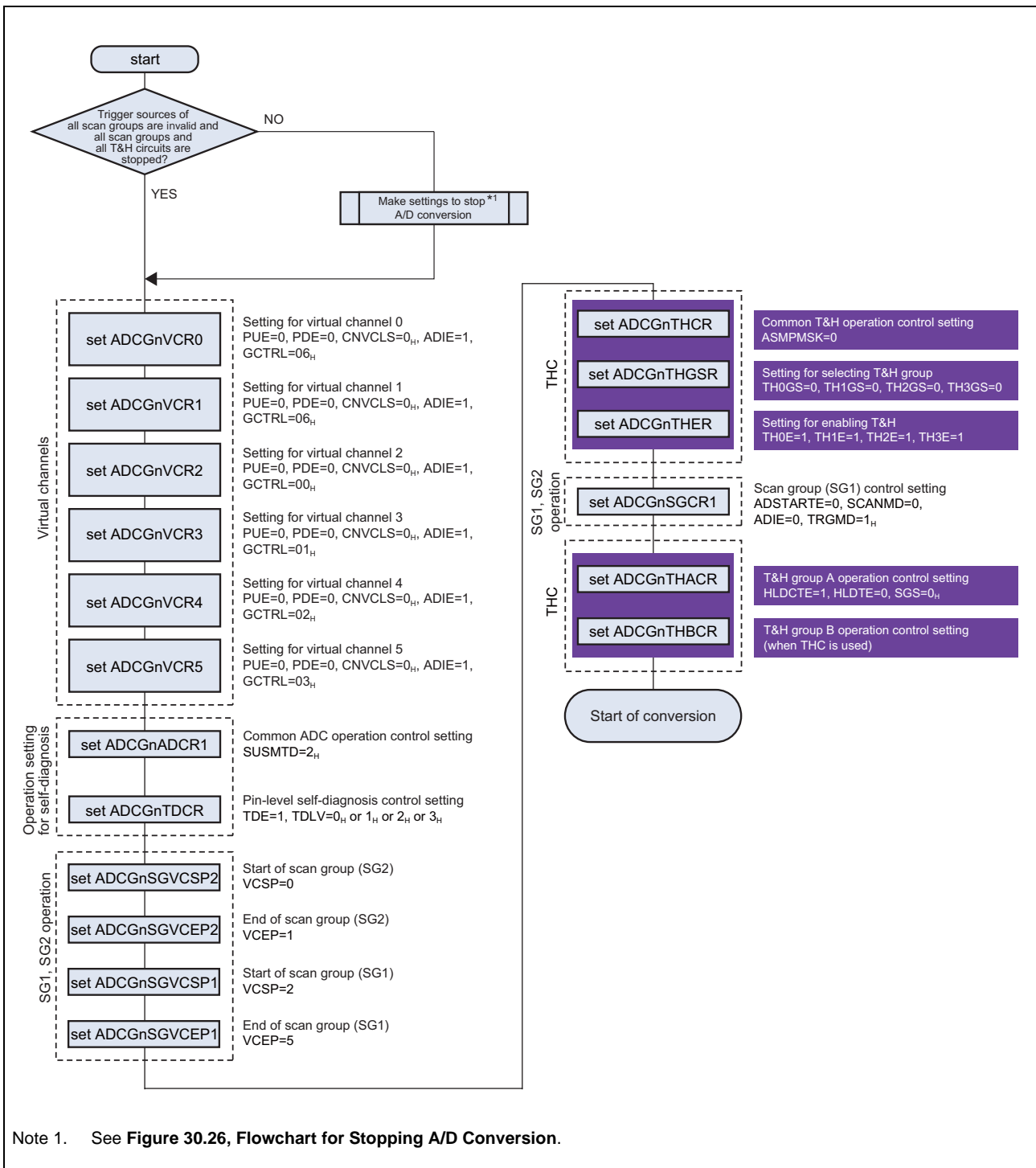


Figure 30.20 Initial Settings for Self-Diagnosis of T&H Paths (the voltage held by the T&H circuits not used)

30.4.5 Interrupt Request Function

There are two interrupt requests: scan group x end and A/D error. An interrupt request signal is a pulse. The scan group x end interrupt can activate the DMA and DTS.

30.4.5.1 Scan End Interrupt Request

Scan group x can issue a scan end interrupt request (INTADCGnIx) to the INTC. When ADIE in ADCGnSGCRx is set to 1, INTADCGnIx is output after the SGx scan ends. When ADIE in ADCGnSGCRx is set to 0, the INTADCGnIx output at the end of the SGx scan is disabled. When ADIE in ADCGnVCRj is set to 1, INTADCGnIx is output when A/D conversion for virtual channel n in SGx ends. When ADIE in ADCGnVCRj is set to 0, the INTADCGnIx output at the end of the A/D conversion for virtual channel n in SGx is disabled. The setting of ADIE in ADCGnSGCRx is independent of the setting of ADIE in ADCGnVCRj.

Example 1) Scans are executed for virtual channel 0 and 1 in SG0 when ADIE in ADCGnSGCR0 is 0, ADIE in ADCGnVCR0 is 1, and ADIE in ADCGnVCR1 is 0. INTADCGnIx is output when A/D conversion ends for virtual channel 0.

Example 2) Scans are executed for virtual channel 0 and 1 in SG0 when ADIE in ADCGnSGCR0 is 0, ADIE in ADCGnVCR0 is 1, and ADIE in ADCGnVCR1 is 1. INTADCGnIx is output when A/D conversion ends for virtual channel 0 and virtual channel 1.

Example 3) Scans are executed for virtual channel 0 and 1 in SG0 when ADIE in ADCGnSGCR0 is 1, ADIE in ADCGnVCR0 is 0, and ADIE in ADCGnVCR1 is 0. INTADCGnIx is output when a scan ends (at the end of A/D conversion for virtual channel 1). INTADCGnIx is output after every two (or more) cycles in multicycle scan mode, or each time a scan ends (at the end of A/D conversion for virtual channel 1) in continuous scan mode.

Furthermore, the DMAC can be activated when INTADCGnIx occurs.

For the DMAC setting, see **Section 7, DMA**.

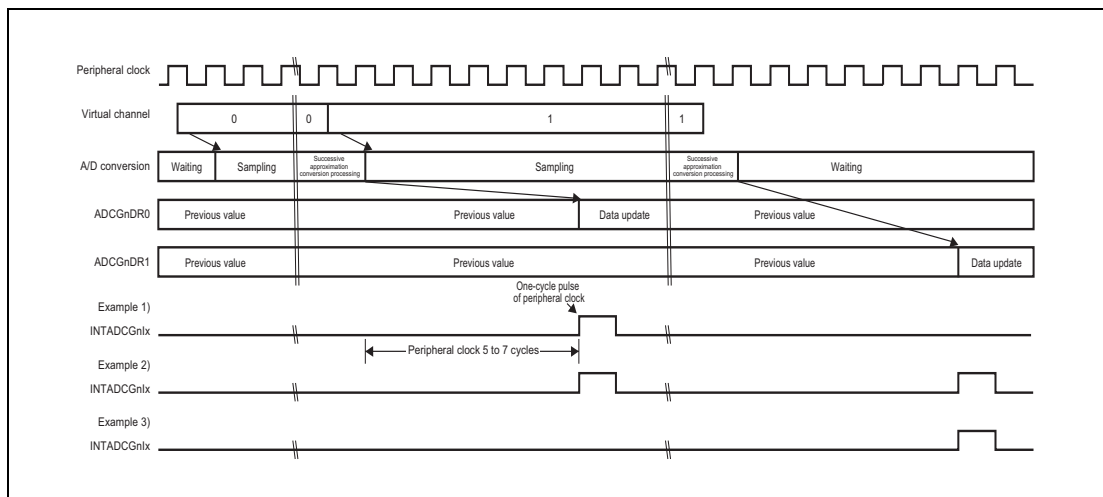


Figure 30.21 Scan Conversion End Interrupt Occurrence Timing

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.4.5.2 A/D Error Interrupt Request and A/D Parity Error Interrupt Request

The ADCG can generate the following error interrupts:

- Upper Limit/Lower Limit Error
- Overwrite Error
- ID Error
- Parity Error

The related registers and the unit to which the interrupts are output differ depending on the source of the error, as shown in **Table 30.53**.

Table 30.53 A/D Error Source

| Error Source | Occurrence condition | Control Register.Bit | Status Register | Clear Register.Bit | Unit Where Error Signal Is Output |
|-------------------------------|---|----------------------|-----------------|--------------------|--|
| Upper Limit/Lower Limit Error | The A/D converted value exceeds the range of the specified upper limit/lower limit table | ADCGnSFTCR. ULEIE | ADCGnULER | ADCGnECCR. ULEC | The OR signal of three error interrupts (AD_ERR) is output to the INTC as the interrupt request INTADCGNERR. |
| Overwrite Error | The A/D converted value is written to ADCGnDRj when ADCGnDIRj.WFLG = 1 | ADCGnSFTCR. OWEIE | ADCGnOWER | ADCGnECCR. OWEC | |
| ID Error | The physical channel specified in ADCGnVCRj does not match the physical channel actually converted. | ADCGnSFTCR. IDEIE | ADCGnIDER | ADCGnECCR. IDEC | |
| Parity Error | A parity error is detected | ADCGnSFTCR. PEIE | ADCGnPER | ADCGnECCR. PEC | A/D parity errors (ACPEn) are output to the ECM. |

The ADCG can issue an A/D error interrupt request (INTADCGnERR) to the INTC and an A/D parity error notification (ADPEn) to the error control module (ECM). For an error source for which ULEIE, OWEIE, and IDEIE in ADCGnSFTCR is set to 1, the OR condition of the error source is issued as INTADCGnERR. For an error source for which ULEIE, OWEIE, and IDEIE in ADCGnSFTCR is set to 0, INTADCGnERR can be disabled. ADPEn is enabled when PEIE in ADCGnSFTCR is set to 1. ADPEn is disabled when PEIE in ADCGnSFTCR is set to 0.

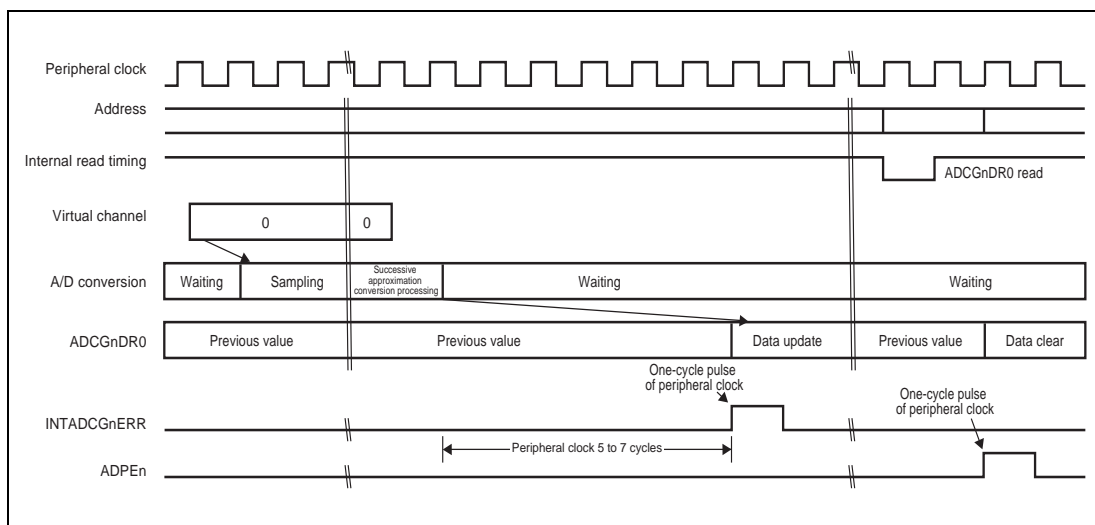


Figure 30.22 Example of an Occurrence of A/D Error Interrupt and A/D Parity Error Notification

NOTE

For the number of units and indices, see **Section 30.1.1, Number of Units**.

30.4.6 Combination of Wiring Break Detection and A/D Conversion for Same Physical Channel

Detection of wiring breaks*1 and A/D conversion with T&H on the same physical channel is supported.

Note 1. In case of using the Wiring-break detection mode 2.

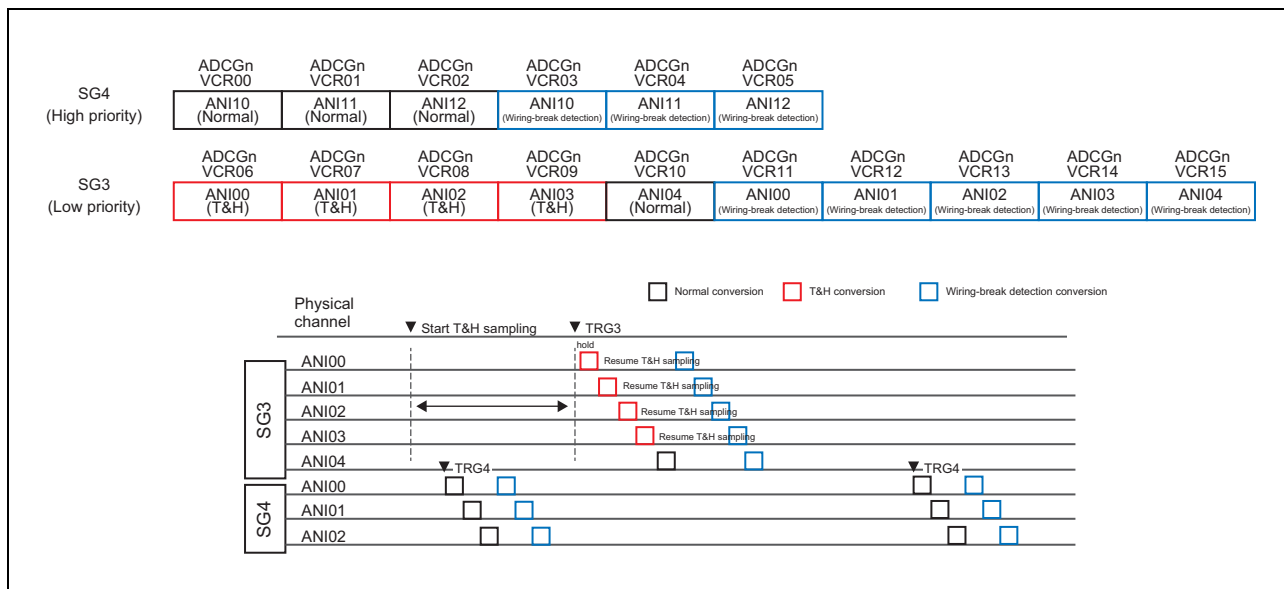


Figure 30.23 Combination of Wiring-Break Detection and A/D Conversion for Same Physical Channel

30.5 Operation

30.5.1 Initial Settings

Figure 30.24 is a flowchart for initial settings. Make initial settings while trigger sources of all scan groups are invalid and all scan groups and all T&H circuits are stopped. If they are in operation, make settings to stop A/D conversion. See **Section 30.5.3, Procedure for Stopping A/D Conversion**.
Set the value after a reset for the functions not to be used.

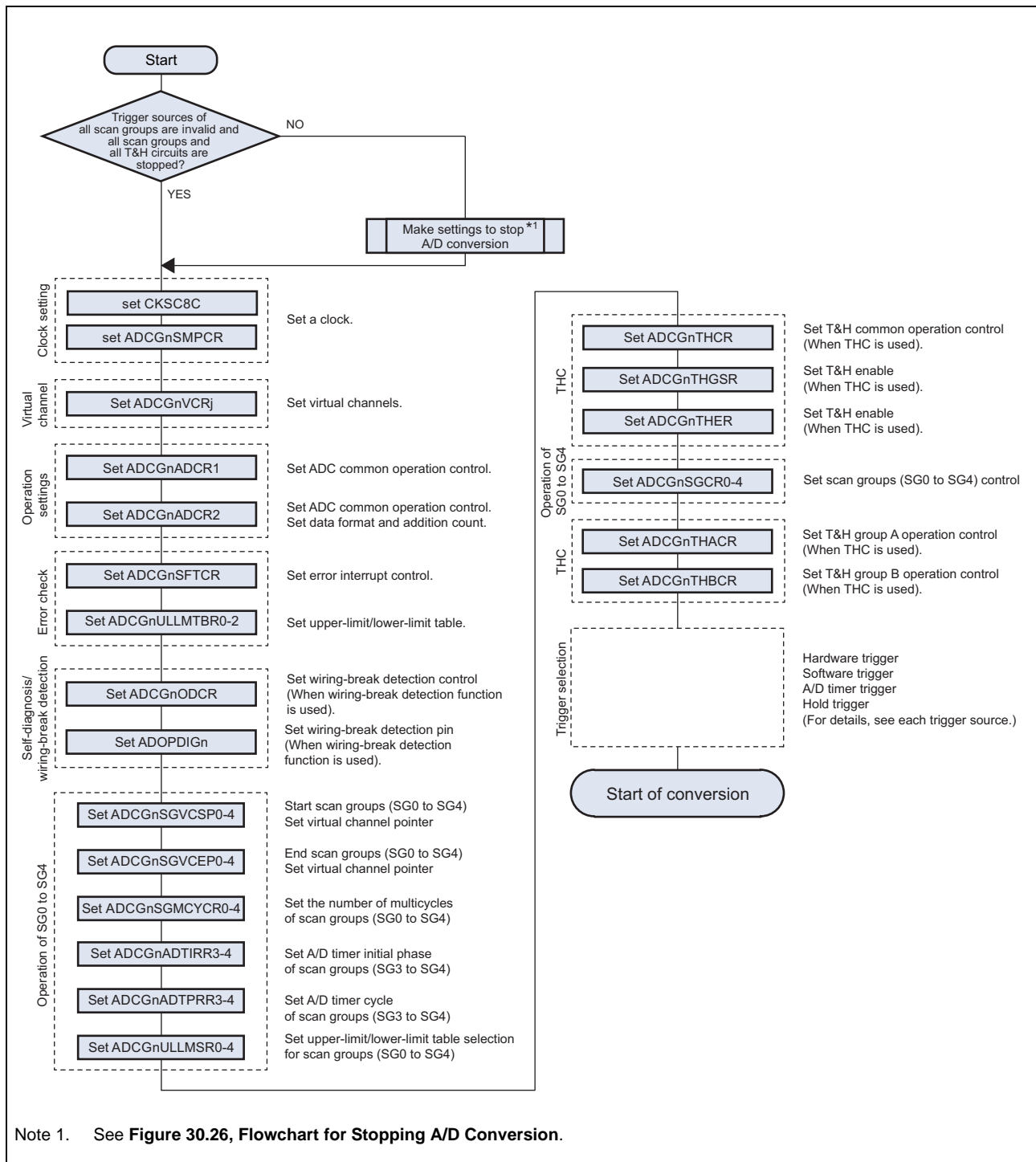


Figure 30.24 Initial Settings

30.5.2 Procedure for Starting A/D Conversion

Figure 30.25 is a flowchart for starting A/D conversion.

The flowchart below is for starting A/D conversion using a hardware trigger when the T&H function is in use. If simultaneous tracking and holding on multiple channels is in use, be sure to only hold the voltage after 32 cycles of CLK_ADC have elapsed after T&H sampling start for tracking and holding.

The flowchart in **Figure 30.25** shows the case where A/D conversion starts under the following conditions.

- Select hardware trigger input to scan group x:
ADCGnSGCRx.TRGMD[1:0] = 1_H
- Automatic sampling of T&H circuits:
ADCGnTHCR.ASMPMSK = 0_H
- Automatic control of T&H hold trigger:
ADCGnTHACR.HLDCTE = 1_H and ADCGnTHACR.HLDTE = 1_H
(or ADCGnTHBCR.HLDCTE=1_H and ADCGnTHBCR.HLDTE=1_H)

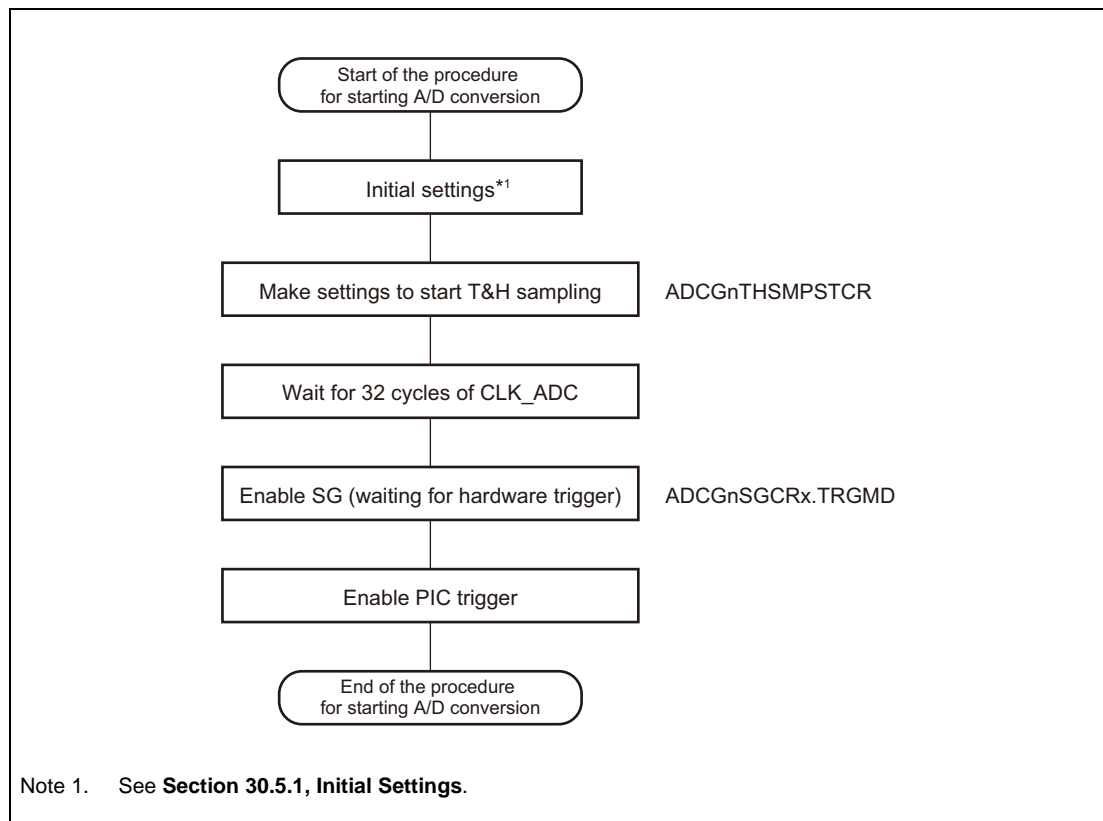


Figure 30.25 Flowchart for Starting A/D Conversion

30.5.3 Procedure for Stopping A/D Conversion

Figure 30.26 is a flowchart for stopping A/D conversion.

The flowchart below is for stopping A/D conversion by disabling the trigger signal for all scan groups and stopping all scan groups and all T&H circuits.

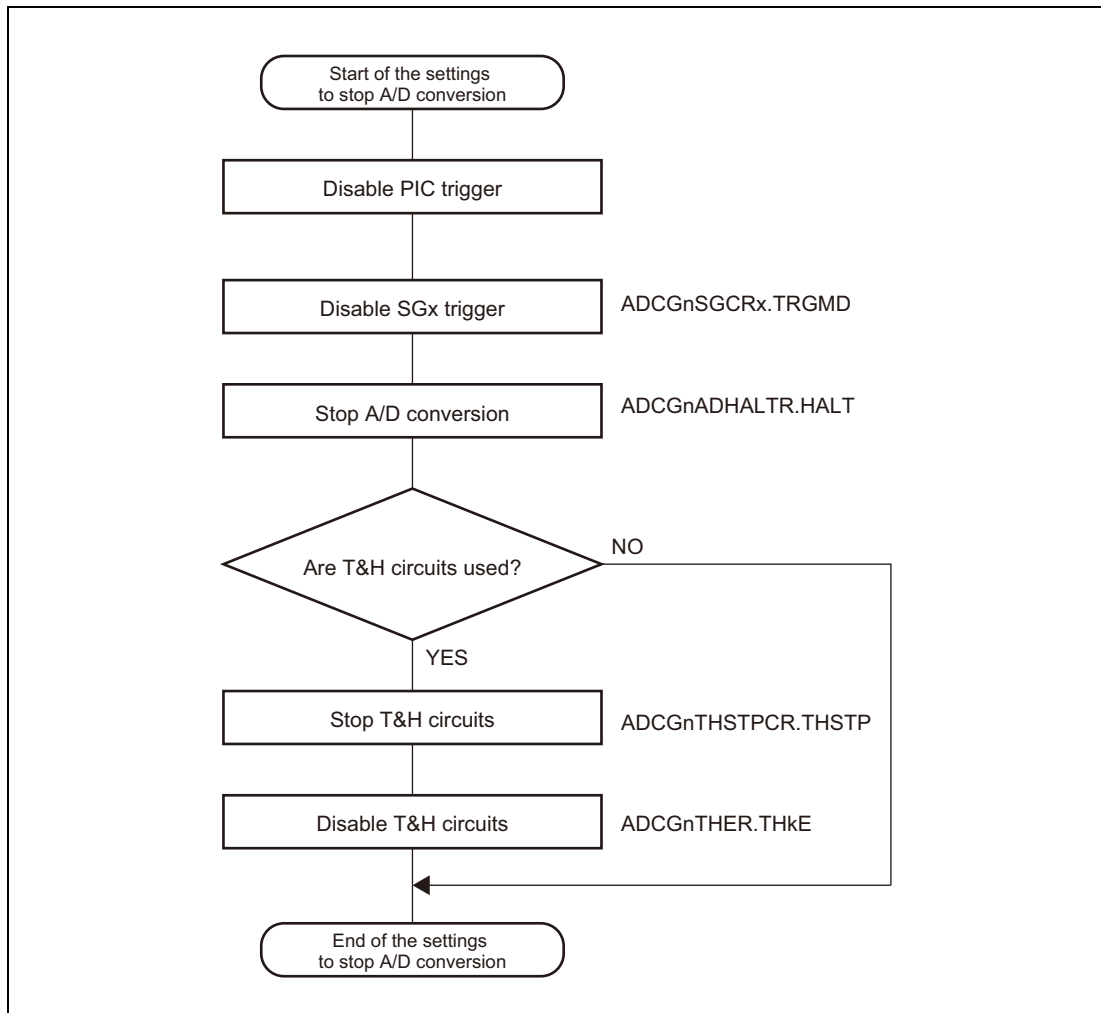


Figure 30.26 Flowchart for Stopping A/D Conversion

30.6 Definition of A/D Conversion Accuracy

A/D conversion accuracy is defined below.

- Resolution
Number of digital output codes of the A/D converter
- Quantization error
An error essentially contained in A/D converters, which is given as 1/2 LSB (**Figure 30.27**).
- Offset error
Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value 000_H to 001_H. The quantization error is not included (**Figure 30.27**).
- Full-scale error
Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from FFE_H to FFF_H. The quantization error is not included (**Figure 30.27**).
- DNL (Differential nonlinear error)
Deviation between the ideal digital output code width (V_q) and the actual digital output code width (V_a), which is given as $(V_a - V_q)/V_q$. The offset error, the full-scale error, and the quantization error are not included (**Figure 30.27**).
- INL (Integral nonlinear error)
Deviation of the actual value from the ideal A/D conversion characteristics, from the zero voltage to the full-scale voltage, which is given as an integral of DNL from 000_H to a digital output code. The offset error, the full-scale error, and the quantization error are not included (**Figure 30.27**).
- Total errors
Maximum difference between the observed value and logical value. The offset error, the full-scale error, the quantization error, DNL, and INL are included (**Figure 30.27**).

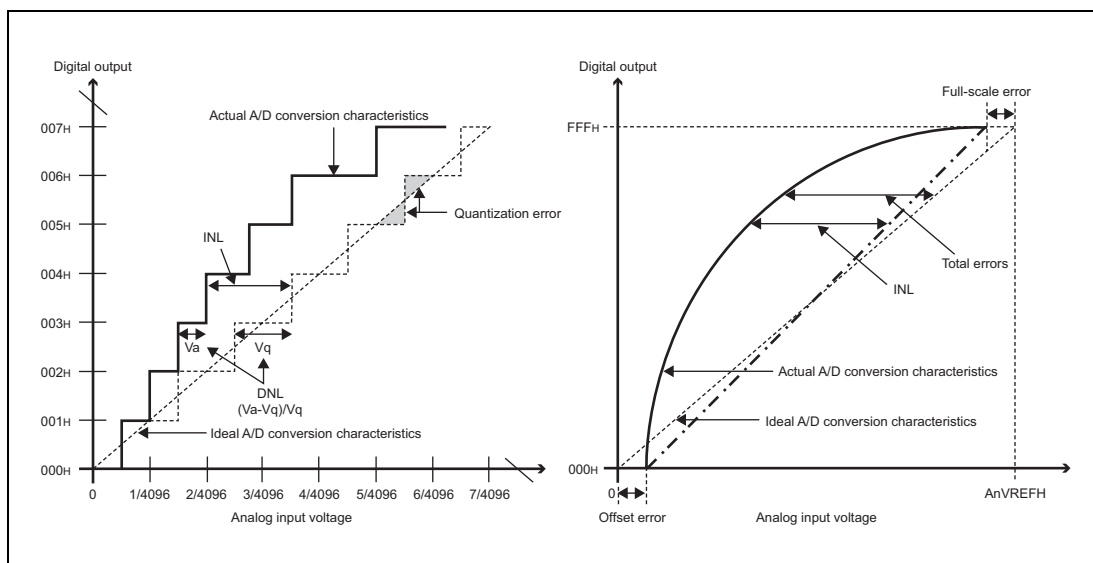


Figure 30.27 Definition of A/D Conversion Accuracy

Section 31 Functional Safety

31.1 Overview

This section describes the failure detection functions provided to detect the LSI failures in the early stage.

31.2 ECC and EDC

31.2.1 Overview

31.2.1.1 ECC

This product incorporates ECC for the following memories. The ECC enables detection and correction of errors of the data retained in the memories. The ECC also enables detection and correction of errors generated between the ECC encoder and memories and between memories and ECC decoder.

Table 31.1 ECC Overview

| Applicable Memory | Applicable Data Width [bits] | Operation upon Error Detection | | | | Failure Insertion |
|--|------------------------------|--------------------------------|------------|--------------|-----------------|-------------------|
| | | Detection/Correction | Notify ECM | Error Status | Address Capture | |
| Code flash | 128 | SEC-DED | Possible | Possible | Possible | Possible |
| Data flash Local RAM (CPU1) Global RAM | 32 | SEC-DED | Possible | Possible | Possible | Possible |
| Instruction cache (data) | 64 | SED-DED | Possible | Possible | Possible | Possible |
| Instruction cache (tag) | 32 | SED-DED | Possible | Possible | Possible | Possible |
| RAM for DTS | 32 | SEC-DED | Possible | Possible | Possible | Possible |
| Peripheral RAM (32 bits) | 32 | SEC-DED | Possible | Possible | Possible | Possible |
| Data transfer path | 32 | SEC-DED | Possible | Possible | Possible | Possible |

ECC code

ECC code with a Hamming distance of minimum 4 is used. Combination of data and ECC code space excludes all 0s or all 1s for RAM and code flash.

Detection/Correction

SEC-DED: 1-bit errors can be detected and corrected, and 2-bit errors can only be detected.

SED-DED: 1-bit errors and 2-bit errors can only be detected.

Notify ECM

The detected error can be notified to the ECM (Error Control Module).

Error status

The status of the detected error is retained.

Address capture

The address of the detected error is retained.

Failure insertion

An ECC error can be intentionally caused to enable self-diagnosis of the ECC decoder operation.

31.2.1.2 Address Parity

This product incorporates address EDC (parity) for the following memories. The address EDC allows detection of errors during address decoding. The EDC also allows detection of errors generated at addresses between the parity encoder and memories.

Table 31.2 Address Parity Overview

| Applicable Memory | Parity Bit | Notify ECM | Error Status | Address Capture | Failure Insertion |
|-------------------|------------|------------|--------------|-----------------|-------------------|
| Code flash | 1 bit | Possible | Possible | Possible | Possible |

31.2.2 Code Flash ECC and Address Parity

31.2.2.1 Overview

The code flash ECC is summarized in the table below.

Table 31.3 Code Flash ECC

| Item | Description |
|------------------------------------|---|
| ECC error detection and correction | <p>ECC error detection and correction can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out.</p> <p>In the initial state, this function is enabled, and 1-bit error detection, correction and notification and 2-bit error detection and notification are carried out.</p> |
| Address parity | <p>Address parity check can be either enabled or disabled.</p> <p>Address parity is checked during address decoding.</p> <p>In the initial state, this function is enabled.</p> |
| Error notification | <p>Upon occurrence of an ECC error or parity error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error.</p> <p>Parity Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an address parity error. <p>In the initial state, error notification is enabled upon detection of an address parity error.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled of an address buffer overflow error for ECC 1-bit error. In the initial state, error notification is enabled upon detection of an address buffer overflow error. <p>The error notification signal is issued to the ECM, in which an ECC 2-bit error and an address parity error are handled as one source, and an ECC 1-bit error is handled as one source, and an ECC 1-bit overflow error is handled as one source. An ECC 1-bit error signal is only issued to the ECM when the detected ECC 1-bit error address is different from the address stored in the error address buffer.</p> |
| Error status | <p>A status register is provided, which indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address parity error detection. A four-stage buffer is provided for 1-bit errors. A one-stage buffer is provided for 2-bit errors and it is shared with parity errors. The status register indicates the state of each stage. The error status can be cleared using the clear register.</p> |
| Address capture | <p>The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address parity error is detected.</p> <p>The error status also serves as the enable bit of the captured address.</p> |

31.2.2.2 List of Registers

Table 31.4 List of Registers

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|------------------------|--------------------|--|-----|------------------------|-------------|
| FFC6 2000 _H | UCFERRINT | Code Flash Error Information Control Register | R/W | 0000 0047 _H | 32/16/8 |
| FFC6 2004 _H | UCFSERSTCLR | Code Flash ECC SED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6 2008 _H | UCFDERSTCLR | Code Flash ECC DED/ Address Parity Error Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6 200C _H | UCFOVFSTR | Code Flash Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC6 2020 _H | UCFSERSTR | Code Flash ECC SED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC6 2030 _H | UCFDERSTR | Code Flash ECC DED/ Address Parity Error Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC6 2040 _H | UCF1SEADR | Code Flash 1 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6 2044 _H | UCF2SEADR | Code Flash 2 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6 2048 _H | UCF3SEADR | Code Flash 3 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6 204C _H | UCF4SEADR | Code Flash 4 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6 20C0 _H | UCFDEADR | Code Flash ECC DED/ Address Parity Error Address Register | R | 0000 0000 _H | 32 |
| FFC6 2100 _H | CFAPCTL | Code Flash Address Parity Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC6 2200 _H | CFECCCTL_VCI2CFBA | Code Flash ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC6 22F0 _H | CFSTSTCTL_VCI2CFBA | Code Flash Sub-Test Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC6 2400 _H | CFECCCTL_PE1 | Code Flash ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC6 24F0 _H | CFSTSTCTL_PE1 | Code Flash Sub-Test Control Register | R/W | 0000 0000 _H | 32/16 |

Note 1. The registers with symbols “_VCI2CFBA” and “_PE1” as suffixes are provided to the particular ECC controllers.

The registers with “_VCI2CFBA” are provided to the ECC controller for access from the system interconnect to the code flash, the registers with “_PE1” are provided to the ECC controller for access from the CPU1.

31.2.2.3 Details of Registers

(1) UCFERRINT — Code Flash Error Information Control Register

UCFERRINT register controls whether error information is reported to ECM, when data ECC 2-bit error, data ECC 1-bit error, and ECC 1-bit error overflow are detected.

Access: UCFERRINT can be read/written in 32/16/8-bit units.

Address: FFC6 2000_H

Value after reset: 0000 0047_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|-------------|----|----|----|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | SEOVFI E | — | — | — | APEIE | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R | R/W | R/W | R/W |

Table 31.5 UCFERRINT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | SEOVFIE | ECC 1bit error overflow report enable bit Controls overflow report when 1bit error overflow flag (SERROVF) in CFOVFSTR register is set. 0: ECC 1-bit error overflow report disabled 1: ECC 1-bit error overflow report enabled |
| 5 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | APEIE | Address parity error report enable bit Controls error report of address parity error detection when address parity check is enabled. 0: Address parity error report disabled 1: Address parity error report enabled |
| 1 | DEDIE | ECC 2-bit error report enable bit Controls error report of 2-bit error detection when ECC error detection/correction is enabled 0:ECC 2-bit error report disabled 1:ECC 2-bit error report enabled |
| 0 | SEDIE | ECC 1-bit error report enable bit Controls error report of 1-bit error detection when ECC error detection/correction is enabled 0:ECC 1-bit error report disabled 1:ECC 1-bit error report enabled |

(2) UCFSERSTCLR — Code Flash ECC SED Status Clear Register

UCFSERSTCLR register is used to clear SEDF[3:0] in UCFSERSTR, SERROVF in UCFOVFSTR, and error address in UCFnSEDADR (n = 1 to 4). This is write only register and read value is always “0”. UCFSERSTCLR register has a lower priority than a set factor. Priority is given to a set factor when UCFSERSTCLR and a set factor compete. A set factor is a trigger for setting SEDF[3:0] in UCFSERSTR, setting SERROVF in UCFOVFSTR, or capturing an error address in UCFnSEDADR (n = 1 to 4).

Access: UCFSERSTCLR can be written only in 32/16/8-bit units.

Address: FFC6 2004_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR3 | SSTCLR2 | SSTCLR1 | SSTCLR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | W | W | W | W |

Table 31.6 UCFSERSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 4 | Reserved | When writing, write the value after reset. |
| 3 to 0 | SSTCLR[3:0] | 1-bit error flag clear (n-th) 0: Nothing operates 1: All 1-bit error flags clear UCFSERSTR.SEDF(n-1), UCFOVFSTR.SERROVF, UCFnSEDADR.SEADR |

(3) UCFDERSTCLR — Code Flash ECC DED/Address Parity Error Status Clear Register

UCFDERSTCLR register is used to clear DEDF and APEF in UCFDERSTR and error address in UCFDEDADR. This is write only register and read value is always “0”. UCFDERSTCLR register has a lower priority than a set factor. Priority is given to a set factor when UCFDERSTCLR and a set factor compete. A set factor is a trigger for setting APEF/DEDF in UCFDERSTR.

Access: UCFSERSTCLR can be written only in 32/16/8-bit units.

Address: FFC6 2008_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.7 UCFDERSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | DSTCLR | 0: Nothing operates 1: All 2-bit error flags clear UCFDERSTR.APEF, UCFDERSTR.DEDF, UCFDEDADR.DEADR |

(4) UCFOVFSTR — Code Flash Error Count Overflow Status Register

UCFOVFSTR register monitors occurrence of error overflow. Overflow occurs when an SED for which the error address has not been captured is detected while ECC 1-bit error status register is full. If the ECC 1-bit error status register is full and the error address is the same as the one of the error addresses already captured, this flag is not set. SERROVF flag is cleared when at least one bit of SSTCLR[3:0] in UCFSERSTCLR register is asserted.

Access: UCFOVFSTR can be read in 32/16/8-bit units.

Address: FFC6 200C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SERR OVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.8 UCFOVFSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | SERROVF | 1-bit error overflow flag When all bits of the ECC 1-bit error status register (UCFSERSTR.SEDF[3:0]) are "1", an ECC 1 bit error occurs and the error address does not match any of the captured error addresses, this flag is set. |

(5) UCFSERSTR — Code Flash ECC SED Status Register

UCFSERSTR is the error monitor register for 1-bit ECC error. Each error flag is “0” and when a new error occurs, an error status flag is set. An error flag is set at the lowest number empty bit of UCFSERSTR (e.g. If SEDF[0][1][3] have been set to “1” and all other bits have been empty, the next flag is set at SEDF[2]). If multiple SED causes are simultaneously detected and there are sufficient empty bits, all detected SEDs are set (e.g. if a SED and another SED which occurs at a different address are detected simultaneously, both SEDFs are set.). However, if SEDs occur at a same address simultaneously, only one of errors is set according to the fixed priority. The priority order is the PE1, and VCI2CFB. (e.g. if a SED which is input from the PE1 path and an SED which is input from the VCI2CFB path are detected simultaneously and those occur at the same address, only SEDF from the PE1 path is set and SEDF from the VCI2CFB path is not set.) Furthermore, an error address which has already been captured in UCFnSEDADR (n = 1 to 4) must not be captured again. UCFSERSTR register is cleared by setting SSTCLR[3:0] in UCFSERSTCLR register to 1.

Access: UCFSERSTR can be read in 32/16/8-bit units.

Address: FFC6 2020_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | SEDF3 | SEDF2 | SEDF1 | SEDF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.9 UCFSERSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 4 | Reserved | When read, the value after reset is read. |
| 3 to 0 | SEDF[3:0] | ECC 1-bit error monitor flag (n-th) Condition for “0”: write “1” to SSTCLR(n-1) in UCFSERSTCLR Condition for “1”: ECC 1-bit error is detected. |

(6) UCFDERSTR — Code Flash ECC DED/Address Parity Error Status Register

UCFDERSTR is the error monitor register for 2-bit ECC error and address parity error. All error flags are “0” and when a new error occurs, an error status flag is set. If multiple error causes which are input from different access ports are detected simultaneously, only one of detected errors is set according to the fixed priority. The reason is that there is only one 2-bit error Address Register. The priority order is the PE1, and VCI2CFB. (e.g. if a DED which is input from the PE1 path and an APE which is input from the VCI2CFB path are detected simultaneously and these occur at the same address, only DEDF from the PE1 path is set and APEF from the VCI2CFB path is not set.) If multiple error causes occur from same access port simultaneously, all detected errors are set. (e.g. if DED and APE are simultaneously input from the PE1 path, both DEDF and APEF are set.) UCFDERSTR register is cleared by setting DSTCLR in UCFDERSTR register to 1.

Access: UCFDERSTR can be read in 32/16/8-bit units.

Address: FFC6 2030_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | APEF | — | DEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.10 UCFDERSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | When read, the value after reset is read. |
| 2 | APEF | Address parity error monitor flag Condition to “0”: write “1” to DSTCLR in UCFDERSTR Condition to “1”: APEF/DEDF is all “0” and address parity error is detected. |
| 1 | Reserved | When read, the value after reset is read. |
| 0 | DEDF | ECC 2-bit error monitor flag Condition to “0”: write “1” to DSTCLR in UCFDERSTR Condition to “1”: DEDF is “0” and ECC 2-bit error is detected. |

(7) UCFnSEDADR — Code Flash n ECC SED Address Register (n = 1 to 4)

UCFnSEDADR registers are used to hold the address when an error is detected. SEADR[24:4] in this register is corresponding to actual address [24:4]. Actual address can be calculated by adding high-order address which is mapped in Code Flash onto base address. Error address capture trigger is same as corresponding SEDFF (n – 1) set in UCFSERSTR. UCFnSEDADR is cleared by setting SSTCLRF (n – 1) in UCFSERSTCLR register to 1.

Access: UCFnSEDADR can be read in 32-bit units.

Address: UCF1SEDADR: FFC6 2040_H
 UCF2SEDADR: FFC6 2044_H
 UCF3SEDADR: FFC6 2048_H
 UCF4SEDADR: FFC6 204C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | SEADR[24:16] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SEADR[15:4] | | | | | | | | | | | | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.11 UCFnSEDADR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 25 | Reserved | When read, the value after reset is read. |
| 24 to 4 | SEADR[24:4] | 1-bit error address |
| 3 to 0 | Reserved | When read, the value after reset is read. |

(8) UCFDEDADR — Code Flash ECC DED/Address Parity Error Address Register

UCFDEDADR register is used to hold the address when an error is detected. DEADR[24:4] in this register is corresponding to actual address [24:4]. Actual address can be calculated by adding high-order address which is mapped in Code Flash onto base address. Error address capture trigger is same as corresponding APEF/DEDF set in UCFDERSTR. If this register captures an error address, it does not capture any more addresses until it is cleared. UCFDEDADR is cleared by setting DSTCLR in UCFDERSTCLR register to 1.

Access: UCFDEDADR can be read in 32-bit units.

Address: FFC6 20C0_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | | | | |
|-------------------|-------------|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|---|---|---|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | | | |
| | — | — | — | — | — | — | — | DEADR[24:16] | | | | | | | | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| | DEADR[15:4] | | | | | | | | | | | | — | — | — | — | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | | | |

Table 31.12 UCFDEDADR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 25 | Reserved | When read, the value after reset is read. |
| 24 to 4 | DEADR[24:4] | 2-bit error or address parity error address |
| 3 to 0 | Reserved | When read, the value after reset is read. |

(9) CFAPCTL — Code Flash Address Parity Control Register

CFAPCTL register enables or disables address parity check. Address parity control register must be written while $PROT[1:0] = 01_B$.

Access: CFAPCTL can be read/written in 32/16-bit units.

Address: FFC6 2100_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | APTES TA | APARID IS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.13 CFAPCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enabled. Other: write is disabled These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | APTESTA | Address Parity Check TEST bit (Bank-A) 0: Address Parity Not Inverted (Normal Mode) 1: Address Parity Inverted (Test Mode) |
| 0 | APARIDIS | Address Parity disable bit Enables or disables address parity check. 0: Address Parity Check is enabled 1: Address Parity Check is disabled |

(10) CFECCTL_VCI2CFBA/PE1 — Code Flash ECC Control Register

CFECCTL_VCI2CFBA/PE1 registers enable or disable the ECC error detection/correction and 1-bit error correction. ECC control registers must be written while $PROT[1:0] = 01_B$.

Access: CFECCTL_VCI2CFBA/PE1 can be read/written in 32/16-bit units.

Address: CFECCTL_VCI2CFBA: FFC6 2200_H
CFECCTL_PE1: FFC6 2400_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|-----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.14 CFECCTL_VCI2CFBA/PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using ECC error detection/correction (ECCDIS = 0). 0: Correction is done when 1-bit error is detected 1: No Correction is done when 1-bit error is detected |
| 0 | ECCDIS | ECC disable bit Enables or disables ECC error detection/correction. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled |

(11) CFSTSTCTL_VCI2CFBA/PE1 — Code Flash Sub-Test Control Register

The CFSTSTCTL registers are used for the ECC test (self-diagnosis). These registers are dedicated for code flash. After ECC test mode is enabled by setting ECCTST = 1, the ECC bits and address parity bit can be read directly.

Access: CFSTSTCTL_VCI2CFBA/PE1 can be read/written in 32/16-bit units.

Address: CFSTSTCTL_VCI2CFBA: FFC6 22F0_H
CFSTSTCTL_PE1: FFC6 24F0_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ECCTST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 31.15 CFSTSTCTL_VCI2CFBA/PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | ECCTST | ECC test bit for Code Flash By setting ECC test mode bit to "1" (ECCTST = 1), CPU can read ECC bits and address parity bit directly. In the read data, bit 9 is the parity bit and bits 0 to 8 are ECC bits. |

Correctly reading instructions from the code flash access port is not possible while ECC test mode is selected (ECCTST = 1). While the access port for the CPU is set to test mode (including during changes to the value of the ECCTST bit), the CPU must run a program from the local RAM and must not fetch instructions from the code flash memory.

The CPU has a small data buffer. If an old value remains in this buffer, the correct value cannot be read even when the ECCTST bit is switched. When switching the ECCTST bit, be sure to clear the data buffer. For how to clear the data buffer, see **Section 3, CPU System**.

From the code flash access port with ECC test mode selected, access must be made by reading 4 bytes aligned to 16n (n: integer) address.

31.2.2.4 Test Function

Through appropriate register setting, the code flash data, ECC bits, and address parity bit can be read out.

(1) Reading code flash data

- (a) Set the ECCDIS bit in the code flash ECC control register to 1 to disable ECC error detection and correction.
- (b) When ECCDIS = 1, neither error detection nor correction is carried out when the code flash is read; the data output from the code flash is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the code flash ECC control register to 0 to enable ECC error detection and correction.

(2) Reading the ECC and address Parity bits

- (a) Set the ECCDIS bit in the code flash ECC control register to 1 to disable ECC error detection and correction.
- (b) Set the ECCTST bit in the code flash sub-test control register to 1 to enter test mode.
- (c) When the code flash is read, the ECC and address parity bits are read instead of the code flash data.

How to exit this test mode:

- (a) Set the ECCDIS bit in the code flash ECC control register to 0 to enable ECC error detection and correction.
- (b) Set the ECCTST bit in the code flash sub-test control register to 0 to return to normal mode.

(3) Self-diagnosis of ECC check function

Self-diagnosis of the ECC decoder for the access ports is possible by reading data from ECC test area.

For details on the ECC test area, refer to **Section 35, Flash Memory**.

(4) Self-diagnosis of address parity check function

- (a) Set the APTESTA bit in the code flash address parity control register to 1 to invert parity bit.
- (b) When code flash data on any address is read, a fault can be injected to the address parity checker to enable self-diagnosis of the address parity check function.

How to exit this test mode:

- (a) Set the APTESTA bit in the code flash address parity control register to 0 to return to normal mode.

31.2.3 Data Flash ECC

31.2.3.1 Overview

The data flash ECC is summarized in the table below.

Table 31.16 Data Flash ECC

| Item | Description |
|------------------------------------|---|
| ECC error detection and correction | <p>ECC error detection and correction can be either enabled or disabled.</p> <p>When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit error detection and correction and 2-bit error detection and notification are carried out.</p> |
| Error notification | <p>Upon occurrence of an ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an address buffer overflow error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error.</p> <p>The error notification signal is output, where an ECC 2-bit error is handled as one source, and an ECC 1-bit error and overflow error are handled as separate sources.</p> |
| Error status | <p>A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.</p> |
| Address capture | <p>If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured.</p> <p>The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status also serves as the enable bit of the captured address.</p> |

31.2.3.2 List of Registers

Table 31.17 List of Registers

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|------------------------|----------|---|-----|------------------------|-------------|
| FFC5 B000 _H | DFECCCTL | Data Flash ECC control register | R/W | 0000 0000 _H | 32/16 |
| FFC5 B004 _H | DFERSTR | Data Flash error status register | R | 0000 0000 _H | 32 |
| FFC5 B008 _H | DFERSTC | Data Flash error status clear register | W | 0000 0000 _H | 32/16/8 |
| FFC5 B00C _H | DFOVFSTR | Data Flash error overflow status register | R | 0000 0000 _H | 32 |
| FFC5 B010 _H | DFOVFSTC | Data Flash error overflow status clear register | W | 0000 0000 _H | 32/16/8 |
| FFC5 B014 _H | DFERRINT | Data Flash error notification control register | R/W | 0000 0006 _H | 32/16/8 |
| FFC5 B018 _H | DFEADR | Data Flash 1st error address register | R | 0000 0000 _H | 32 |
| FFC5 B01C _H | DFTSTCTL | Data flash test control register | R/W | 0000 0000 _H | 32/16 |

31.2.3.3 Details of Registers

(1) DFECCTL — Data Flash ECC Control Register

DFECCTL enables or disables ECC error detection and 1-bit error correction for read access to each data flash bank. Set the PROT1 and PROT0 bits to 01_B when writing to DFECCTL.

Access: DFECCTL can be read/written in 32/16-bit units.

Address: FFC5 B000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.18 DFECCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 | PROT1 | Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFECCTL. |
| 14 | PROT0 | |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | 1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Set (PROT1, PROT0) = (0, 1) simultaneously when writing to this bit. 0: Enables correction of the 1-bit error detected. 1: Disables correction of the 1-bit error detected. |
| 0 | ECCDIS | ECC Disable Enables or disables ECC error detection and correction. Set (PROT1, PROT0) = (0, 1) simultaneously when writing to this bit. In the initial state, ECC error detection and correction are enabled. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction. |

(2) DFERSTR — Data Flash Error Status Register

DFERSTR monitors occurrence of errors.

The SEDF bit is set if an ECC 1-bit error is detected and the DEDF bit is set if an ECC 2-bit error is detected while ECC error detection and correction are enabled.

Access: DFERSTR can be read only in 32-bit units.

Address: FFC5 B004_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDF | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.19 DFERSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF | ECC 2-bit Error Monitor Flag 0: An ECC 2-bit error is not generated. 1: An ECC 2-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 2-bit error is generated. |
| 0 | SEDF | ECC 1-bit Error Monitor Flag 0: An ECC 1-bit error is not generated. 1: An ECC 1-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 1-bit error is generated with both SEDF and DEDF being 0. |

(3) DFERSTC — Data Flash Error Status Clear Register

DFERSTC clears the error flags in the Data Flash error status register.

Access: DFERSTC can be written only in 32/16/8-bit units.

Address: FFC5 B008_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERRCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.20 DFERSTC Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ERRCLR | SEDF/DEDF Flag Clear 0: No effect (Setting the ERRCLR bit to 0 does not affect the DEDF and SEDF flags in DFERSTR.) 1: The SEDF/DEDF flags in DFERSTR are cleared. |

(4) DFOVFSTR — Data Flash Error Overflow Status Register

DFOVFSTR monitors occurrence of Data Flash error overflow.

Access: DFOVFSTR can be read only in 32-bit units.

Address: FFC5 B00C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERR OVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.21 DFOVFSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error Overflow Flag ERROVF is set as follows: <ul style="list-style-type: none"> • An ECC 1-bit error has occurred when DFERSTR.SEDF = 1 and access address is different from DFEADR. • An ECC 1-bit error has occurred when DFERSTR.DEDF = 1 • An ECC 2-bit error has occurred when DFERSTR.SEDF = 1 • An ECC 2-bit error has occurred when DFERSTR.DEDF = 1 and access address is different from DFEADR. 0: Not occurred 1: Occurred |

Clearing condition: Set the ERROVFCLR bit in DFOVFSTC to 1.

(5) DFOVFSTC — Data Flash Error Overflow Status Clear Register

DFOVFSTC clears the data flash error overflow flag.

Access: DFOVFSTC can be written only in 32/16/8-bit units.

Address: FFC5 B010_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERR OVF CLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.22 DFOVFSTC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ERROVFCLR | Error Overflow Flag Clear 0: No effect (Setting the ERROVFCLR bit to 0 does not affect the ERROVF flag in DFOVFSTR.) 1: The ERROVF flag in the DFOVFSTR register is cleared. |

(6) DFERRINT — Data Flash Error Notification Control Register

DFERRINT enables or disables generation of the error notification signal to ECM upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access: DFERRINT can be read/written in 32/16/8-bit units.

Address: FFC5 B014_H

Value after reset: 0000 0006_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | EOVFIE | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W |

Table 31.23 DFERRINT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 3 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 2 | EOVFIE | ECC Error Address Overflow Notification Control Enables or disables generation of the error notification signal upon detection of an address buffer overflow. 0: Disables notification of error address overflow 1: Enables notification of error address overflow |
| 1 | DEDIE | ECC 2-bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error. |
| 0 | SEDIE | ECC 1-bit Error Notification Control Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error. |

(7) DFEADR — Data Flash 1st Error Address Register

DFEADR holds the error address if the following occur.

- An ECC 1-bit or 2-bit error occurred when DFERSTR.SEDF = 0 and DFERSTR.DEDF = 0
- An ECC 2-bit error occurred when DFERSTR.SEDF = 1 and DFERSTR.DEDF = 0

Access: This register can be read only in 32-bit units.

Address: FFC5 B018_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | DFEADR[20:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DFEADR[15:2] | | | | | | | | | | | | | | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.24 DFEADR Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. |
| 20 to 2 | DFEADR[20:2] | ECC Error Address DFEADR is a read-only field to monitor the address at which an ECC error has occurred. This register holds an internal address. Convert it to the actual address by adding the data flash base address FF20 0000 _H . |
| 1, 0 | Reserved | When read, the value after reset is read. |

(8) DFTSTCTL — Data Flash Test Control Register

DFTSTCTL is used for ECC testing.

The data of the ECC bit can be read after setting the ECC test mode (ECCTST = 1).

This register must be written while PROT[1:0] = 01_B.

Access: DFTSTCTL can be read/written in 32/16-bit units.

Address: FFC5 B01C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | — | ECC TST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 31.25 DFTSTCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 | PROT1 | Enables or disables modification of the ECCTST bit. |
| 14 | PROT0 | The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFTSTCTL. |
| 13 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | ECCTST | ECC Test By setting ECC Test bit to "1" (ECCTST=1), CPU can read ECC bit. Set (PROT1, PROT0) = (0, 1) simultaneously when writing to this bit. |

31.2.3.4 Test Function

Data in the ROM and the ECC bits can be read by setting the data flash test control register (DFTSTCTL).

(1) Reading the ROM Data

- (a) Set the ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.
- (b) When ECCDIS = 1, neither error detection nor correction is carried out when the data flash is read; the data output from the data flash is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the data flash ECC control register to 0 to enable ECC error detection and correction.

(2) Reading the ECC bits

- (a) Set the ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.
- (b) Set the ECCTST bit in the data flash test control register to 1 to enter test mode.
- (c) When the data flash is read, the 7 lower-order bits of read data are read as ECC data.

How to exit this test mode:

- (a) Set the ECCDIS bit in the data flash ECC control register to 0 to enable ECC error detection and correction.
- (b) Set the ECCTST bit in the data flash test control register to 0 to return to normal mode.

(3) Self-diagnosis of ECC check function

Self-diagnosis of the ECC decoder is possible by writing incorrect data to the data flash memory beforehand (fault injection) and then reading this data. A 1- or 2-bit ECC error fault can be injected by generating correct ECC bits once and inverting only the appropriate bits.

For details on programming of the data flash, refer to *RH850/P1M-E Flash Memory User's Manual: Hardware Interface*.

31.2.4 Local RAM (CPU1) ECC

31.2.4.1 Overview

Local RAM ECC of CPU1 is summarized in the table below.

Table 31.26 Local RAM ECC

| Item | Description |
|------------------------------------|---|
| ECC error detection and correction | <p>ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled; 1-bit error detection, correction, and notification, and 2-bit error detection and notification are carried out.</p> |
| Error notification | <p>Upon occurrence of an ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, notification of the 2-bit error is enabled and notification of the 1-bit error is enabled.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error.</p> <p>The error notification signal is output, where an ECC 2-bit error is handled as one source, an ECC 1-bit error is handled as one source, and an ECC 1-bit overflow error is handled as one source.</p> <p>An ECC1-bit error signal is only issued to the ECM when the detected ECC 1-bit error address is different from the address stored in the error address buffer.</p> |
| Error status | <p>A status register is provided, which indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.</p> |
| Address capture | <p>Multi-stage address buffers are provided for ECC 1bit errors as with code flash ECC.</p> <p>1-bit error: Eight stages (in 32-bit units) 2-bit error: One stage (in 32-bit units)</p> <p>The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status also serves as the enable bit of the captured address.</p> |

CPU1 is capable of simultaneously writing or reading up to 128 bits of data at a time to or from the local RAM. Meanwhile, ECC bits are provided for each 32 bits of data and the locations for storage of each 32-bit data segment are referred to as banks 0 to 3. Addresses for the lowest-order bits (i.e., for the LSB side of the data) are for bank 0, while addresses for the highest-order bits (i.e., for the MSB side of the data) are for bank 3.

The relationship between addresses and banks is described below.

| 4 Lower-Order Bits of Address (Hexadecimal Notation) | F _H to C _H | B _H to 8 _H | 7 _H to 4 _H | 3 _H to 0 _H |
|---|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Bank number | Bank 3 | Bank 2 | Bank 1 | Bank 0 |

31.2.4.2 List of Registers

Table 31.27 List of Registers (1/2)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|----------------|--|-----|------------------------|-------------|
| FFC65400 _H | LRECCCTL_PE1 | Local-RAM ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC65404 _H | LRERRINT_PE1 | Local RAM error information control register | R/W | 0000 0043 _H | 32/16/8 |
| FFC65408 _H | LRSERSTCLR_PE1 | Local-RAM ECC SED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6540C _H | LRDERSTCLR_PE1 | Local-RAM ECC DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC65410 _H | LROVFSTR_PE1 | Local RAM error count overflow status register | R | 0000 0000 _H | 32/16/8 |
| FFC65440 _H | LRSERSTR_PE1 | Local-RAM ECC SED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC65450 _H | LRDERSTR_PE1 | Local-RAM ECC DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC65460 _H | LR1SEDADR0_PE1 | Local RAM 1st error address register 0 | R | 0000 0000 _H | 32 |
| FFC65464 _H | LR1SEDADR1_PE1 | Local RAM 1st error address register 1 | R | 0000 0000 _H | 32 |
| FFC65468 _H | LR1SEDADR2_PE1 | Local RAM 1st error address register 2 | R | 0000 0000 _H | 32 |
| FFC6546C _H | LR1SEDADR3_PE1 | Local RAM 1st error address register 3 | R | 0000 0000 _H | 32 |
| FFC65470 _H | LR2SEDADR0_PE1 | Local RAM 2nd error address register 0 | R | 0000 0000 _H | 32 |
| FFC65474 _H | LR2SEDADR1_PE1 | Local RAM 2nd error address register 1 | R | 0000 0000 _H | 32 |
| FFC65478 _H | LR2SEDADR2_PE1 | Local RAM 2nd error address register 2 | R | 0000 0000 _H | 32 |
| FFC6547C _H | LR2SEDADR3_PE1 | Local RAM 2nd error address register 3 | R | 0000 0000 _H | 32 |
| FFC65480 _H | LR3SEDADR0_PE1 | Local RAM 3rd error address register 0 | R | 0000 0000 _H | 32 |
| FFC65484 _H | LR3SEDADR1_PE1 | Local RAM 3rd error address register 1 | R | 0000 0000 _H | 32 |
| FFC65488 _H | LR3SEDADR2_PE1 | Local RAM 3rd error address register 2 | R | 0000 0000 _H | 32 |
| FFC6548C _H | LR3SEDADR3_PE1 | Local RAM 3rd error address register 3 | R | 0000 0000 _H | 32 |
| FFC65490 _H | LR4SEDADR0_PE1 | Local RAM 4th error address register 0 | R | 0000 0000 _H | 32 |
| FFC65494 _H | LR4SEDADR1_PE1 | Local RAM 4th error address register 1 | R | 0000 0000 _H | 32 |
| FFC65498 _H | LR4SEDADR2_PE1 | Local RAM 4th error address register 2 | R | 0000 0000 _H | 32 |
| FFC6549C _H | LR4SEDADR3_PE1 | Local RAM 4th error address register 3 | R | 0000 0000 _H | 32 |
| FFC654A0 _H | LR5SEDADR0_PE1 | Local RAM 5th error address register 0 | R | 0000 0000 _H | 32 |

Table 31.27 List of Registers (2/2)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|----------------|--|-----|------------------------|-------------|
| FFC654A4 _H | LR5SEDADR1_PE1 | Local RAM 5th error address register 1 | R | 0000 0000 _H | 32 |
| FFC654A8 _H | LR5SEDADR2_PE1 | Local RAM 5th error address register 2 | R | 0000 0000 _H | 32 |
| FFC654AC _H | LR5SEDADR3_PE1 | Local RAM 5th error address register 3 | R | 0000 0000 _H | 32 |
| FFC654B0 _H | LR6SEDADR0_PE1 | Local RAM 6th error address register 0 | R | 0000 0000 _H | 32 |
| FFC654B4 _H | LR6SEDADR1_PE1 | Local RAM 6th error address register 1 | R | 0000 0000 _H | 32 |
| FFC654B8 _H | LR6SEDADR2_PE1 | Local RAM 6th error address register 2 | R | 0000 0000 _H | 32 |
| FFC654BC _H | LR6SEDADR3_PE1 | Local RAM 6th error address register 3 | R | 0000 0000 _H | 32 |
| FFC654C0 _H | LR7SEDADR0_PE1 | Local RAM 7th error address register 0 | R | 0000 0000 _H | 32 |
| FFC654C4 _H | LR7SEDADR1_PE1 | Local RAM 7th error address register 1 | R | 0000 0000 _H | 32 |
| FFC654C8 _H | LR7SEDADR2_PE1 | Local RAM 7th error address register 2 | R | 0000 0000 _H | 32 |
| FFC654CC _H | LR7SEDADR3_PE1 | Local RAM 7th error address register 3 | R | 0000 0000 _H | 32 |
| FFC654D0 _H | LR8SEDADR0_PE1 | Local RAM 8th error address register 0 | R | 0000 0000 _H | 32 |
| FFC654D4 _H | LR8SEDADR1_PE1 | Local RAM 8th error address register 1 | R | 0000 0000 _H | 32 |
| FFC654D8 _H | LR8SEDADR2_PE1 | Local RAM 8th error address register 2 | R | 0000 0000 _H | 32 |
| FFC654DC _H | LR8SEDADR3_PE1 | Local RAM 8th error address register 3 | R | 0000 0000 _H | 32 |
| FFC654E0 _H | LRDEDADR0_PE1 | Local-RAM DED Address Register 0 | R | 0000 0000 _H | 32 |
| FFC654E4 _H | LRDEDADR1_PE1 | Local-RAM DED Address Register 1 | R | 0000 0000 _H | 32 |
| FFC654E8 _H | LRDEDADR2_PE1 | Local-RAM DED Address Register 2 | R | 0000 0000 _H | 32 |
| FFC654EC _H | LRDEDADR3_PE1 | Local-RAM DED Address Register 3 | R | 0000 0000 _H | 32 |
| FFC65604 _H | LRTSTCTL_PE1 | Local-RAM Test Control Register | R/W | 0000 0000 _H | 32 |
| FFC65608 _H | LRTDATBF0_PE1 | Local-RAM Test Data Read Buffer 0 | R | 0000 0000 _H | 32 |
| FFC6560C _H | LRTDATBF1_PE1 | Local-RAM Test Data Read Buffer 1 | R | 0000 0000 _H | 32 |

31.2.4.3 Details of Registers

(1) LRECCCTL_PE1 — Local-RAM ECC Control Register

LRECCCTL_PE1 register controls the ECC error detection/correction and 1-bit error correction.
Writing ECC control registers must be executed with PROT[1:0] = 01_B.

Access: LRECCCTL_PE1 can be read/written in 32/16-bit units.

Address: FFC6 5400_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.28 LRECCCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using ECC error detection/correction (ECCDIS = 0). 0: Correction is carried out when 1-bit error is detected 1: Correction is not carried out when 1-bit error is detected |
| 0 | ECCDIS | ECC disable bit Enables or disables ECC error detection/correction. 0: ECC error detection/correction is enable 1: ECC error detection/correction is disable |

(2) LRERRINT_PE1 — Local-RAM Error Information Control Register

LRERRINT_PE1 register controls whether error information is reported to ECM, when data ECC 2-bit error, data ECC 1-bit error, and ECC 1-bit error overflow are detected.

Access: LRERRINT_PE1 can be read/written in 32/16/8-bit units.

Address: FFC6 5404_H

Value after reset: 0000 0043_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | EOVFIE | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R | R | R/W | R/W |

Table 31.29 LRERRINT_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | EOVFIE | ECC 1bit error overflow report enable bit Controls overflow report when 1bit error overflow flag (SERROVFn) in LROVFSTR register is set. 0: ECC 1bit error overflow report disabled 1: ECC 1bit error overflow report enabled |
| 5 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | ECC 2-bit error report enable bit Controls error report of 2-bit error detection when ECC error detection/correction is enabled 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled |
| 0 | SEDIE | ECC 1-bit error report enable bit Controls error report of 1-bit error detection when ECC error detection/correction is enabled 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled |

(3) LRSERSTCLR_PE1— Local-RAM ECC SED Status Clear Register

LRSERSTCLR_PE1 register is used to clear SEDF_{mn} (m = 1 to 8, n = 0 to 3) in LRSERSTR_PE1 and SERROV_{Fn} (n = 0 to 3) in LROVFSTR_PE1, and error address in LR[m]SEDADR[n]_PE1. This is write only register and read value is always “0”.

Access: LRSERSTCLR_PE1 can be written in 32/16/8-bit units.

Address: FFC6 5408_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | SSTCL R83 | SSTCL R82 | SSTCL R81 | SSTCL R80 | SSTCL R73 | SSTCL R72 | SSTCL R71 | SSTCL R70 | SSTCL R63 | SSTCL R62 | SSTCL R61 | SSTCL R60 | SSTCL R53 | SSTCL R52 | SSTCL R51 | SSTCL R50 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SSTCL R43 | SSTCL R42 | SSTCL R41 | SSTCL R40 | SSTCL R33 | SSTCL R32 | SSTCL R31 | SSTCL R30 | SSTCL R23 | SSTCL R22 | SSTCL R21 | SSTCL R20 | SSTCL R13 | SSTCL R12 | SSTCL R11 | SSTCL R10 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 31.30 LRSERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--|--|
| 31 to 0 | SSTCLR _{mn} (m = 1 to 8) (n = 0 to 3) | 1-bit error flag clear (bank n) 0: Nothing operates 1: All 1-bit error flags clear LRSERSTR_PE1.SEDF _{mn} , LROVFSTR_PE1.SERROV _{Fn} , LR[m]SEDADR[n]_PE1.SEADR |

(4) LRDERSTCLR_PE1 — Local-RAM ECC DED Status Clear Register

LRDERSTCLR_PE1 register is used to clear DEDFn (n = 0 to 3) in LRDERSTR_PE1 and error address in LRDEDADRn_PE1 (n = 0 to 3). This is write only register and read value is always “0”.

Access: LRDERSTCLR_PE1 can be written in 32/16/8-bit units.

Address: FFC6 540C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-------------|-------------|-------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | DSTCL R3 | DSTCL R2 | DSTCL R1 | DSTCL R0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | W | W | W | W |

Table 31.31 LRDERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 4 | Reserved | When writing, write the value after reset. |
| 3 to 0 | DSTCLR[3:0] | 2-bit error flag clear (bank n) 1: All 2-bit error flags clear LRDERSTR_PE1.DEDFn, LRDEDADRn_PE1.DEADR |

(5) LROVFSTR_PE1 — Local-RAM Error Count Overflow Status Register

LROVFSTR_PE1 register monitors occurrence of error overflow. ECC 1bit error occurs at same error address is detected while the error status is full. If the error status is full and the same error (same error address) has occurred, then this flag is not set. SERROVFn (n = 0 to 3) flag is cleared by setting SSTCLRmn (m = 1 to 8, n = 0 to 3) in LRSERSTCLR_PE1 register to 1.

Access: LROVFSTR_PE1 can be read in 32/16/8-bit units.

Address: FFC6 5410_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|--------------|--------------|--------------|--------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | SERRO VF3 | SERRO VF2 | SERRO VF1 | SERRO VF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.32 LROVFSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|--|
| 31 to 4 | Reserved | When read, the value after reset is read. |
| 3 to 0 | SERROVF[3:0] | 1-bit error overflow flag (bank n) When all bits provided each bank of the ECC 1-bit error status register (LRSERSTR.SEDF[8:1]n) are "1", an ECC 1 bit error occurs and the error address does not match any of the captured error addresses, this flag is set. |

(6) LRSERSTR_PE1 — Local-RAM ECC SED Status Register

LRSERSTR_PE1 is the error monitor register. If the error flag is “0” for each bank and a new error occurs, the error status flag is set. LRSERSTR_PE1 register is cleared by setting SSTCLRmn (m = 1 to 8, n = 0 to 3) in LRSERSTCLR_PE1 register to 1.

Access: LRSERSTR_PE1 can be read in 32/16/8-bit units.

Address: FFC6 5440_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | SEDF8 3 | SEDF8 2 | SEDF8 1 | SEDF8 0 | SEDF7 3 | SEDF7 2 | SEDF7 1 | SEDF7 0 | SEDF6 3 | SEDF6 2 | SEDF6 1 | SEDF6 0 | SEDF5 3 | SEDF5 2 | SEDF5 1 | SEDF5 0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SEDF4 3 | SEDF4 2 | SEDF4 1 | SEDF4 0 | SEDF3 3 | SEDF3 2 | SEDF3 1 | SEDF3 0 | SEDF2 3 | SEDF2 2 | SEDF2 1 | SEDF2 0 | SEDF1 3 | SEDF1 2 | SEDF1 1 | SEDF1 0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.33 LRSERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--|--|
| 31 to 0 | SEDFmn (m = 1 to 8) (n = 0 to 3) | ECC 1-bit error monitor flag (bank n) Condition to “0”: write “1” to SSTCLRmn in LRSERSTCLR_PE1 Condition to “1”: SEDFmn is “0” and ECC 1-bit error is detected. |

(7) LRDERSTR_PE1 — Local-RAM ECC DED Status Register

LRDERSTR_PE1 is the error monitor register. If the error flag is “0” for each bank and a new ECC 2-bit error occurs by reading from this bank, then error status flag is set. LRDERSTR_PE1 register is cleared by setting DSTCLRn (n = 0 to 3) in LRDERSTCLR_PE1 register to 1.

Access: LRDERSTR_PE1 can be read in 32/16/8-bit units.

Address: FFC6 5450_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|----|
| | — | — | — | — | — | — | DEDF3 | — | — | — | — | — | — | — | DEDF2 | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | DEDF1 | — | — | — | — | — | — | — | DEDF0 | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.34 LRDERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 26 | Reserved | When read, the value after reset is read. |
| 25 | DEDF3 | ECC 2-bit error monitor flag (bank 3) Condition to “0”: write “1” to DSTCLR3 in LRDERSTCLR_PE1 Condition to “1”: DEDF3 is “0” and ECC 2-bit error is detected. |
| 24 to 18 | Reserved | When read, the value after reset is read. |
| 17 | DEDF2 | ECC 2-bit error monitor flag (bank 2) Condition to “0”: write “1” to DSTCLR2 in LRDERSTCLR_PE1 Condition to “1”: DEDF2 is “0” and ECC 2-bit error is detected. |
| 16 to 10 | Reserved | When read, the value after reset is read. |
| 9 | DEDF1 | ECC 2-bit error monitor flag (bank 1) Condition to “0”: write “1” to DSTCLR1 in LRDERSTCLR_PE1 Condition to “1”: DEDF1 is “0” and ECC 2-bit error is detected. |
| 8 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF0 | ECC 2-bit error monitor flag (bank 0) Condition to “0”: write “1” to DSTCLR0 in LRDERSTCLR_PE1 Condition to “1”: DEDF0 is “0” and ECC 2-bit error is detected. |
| 0 | Reserved | These bits are always read as 0. The write value should also be 0. |

(8) LRmSEDADRn_PE1 — Local-RAM 1st to 8th SED Address Register n (m = 1 to 8, n = 0 to 3)

LRmSEDADRn_PE1 (m = 1 to 8, n = 0 to 3) registers are used to hold the address when an error is detected. Error address capture trigger is same as corresponding SEDFmn (m = 1 to 8, n = 0 to 3) set in LRSERSTR_PE1. LRmSEDADRn_PE1 (m = 1 to 8, n = 0 to 3) are cleared by setting SSTCLRmn (m = 1 to 8, n = 0 to 3) in LRSERSTCLR_PE1 register to 1.

Access: LRmSEDADRn_PE1 can be read in 32-bit units.

Address:

| | |
|--|--|
| LR1SEDADR0_PE1: FFC6 5460 _H | LR1SEDADR1_PE1: FFC6 5464 _H |
| LR1SEDADR2_PE1: FFC6 5468 _H | LR1SEDADR3_PE1: FFC6 546C _H |
| LR2SEDADR0_PE1: FFC6 5470 _H | LR2SEDADR1_PE1: FFC6 5474 _H |
| LR2SEDADR2_PE1: FFC6 5478 _H | LR2SEDADR3_PE1: FFC6 547C _H |
| LR3SEDADR0_PE1: FFC6 5480 _H | LR3SEDADR1_PE1: FFC6 5484 _H |
| LR3SEDADR2_PE1: FFC6 5488 _H | LR3SEDADR3_PE1: FFC6 548C _H |
| LR4SEDADR0_PE1: FFC6 5490 _H | LR4SEDADR1_PE1: FFC6 5494 _H |
| LR4SEDADR2_PE1: FFC6 5498 _H | LR4SEDADR3_PE1: FFC6 549C _H |
| LR5SEDADR0_PE1: FFC6 54A0 _H | LR5SEDADR1_PE1: FFC6 54A4 _H |
| LR5SEDADR2_PE1: FFC6 54A8 _H | LR5SEDADR3_PE1: FFC6 54AC _H |
| LR6SEDADR0_PE1: FFC6 54B0 _H | LR6SEDADR1_PE1: FFC6 54B4 _H |
| LR6SEDADR2_PE1: FFC6 54B8 _H | LR6SEDADR3_PE1: FFC6 54BC _H |
| LR7SEDADR0_PE1: FFC6 54C0 _H | LR7SEDADR1_PE1: FFC6 54C4 _H |
| LR7SEDADR2_PE1: FFC6 54C8 _H | LR7SEDADR3_PE1: FFC6 54CC _H |
| LR8SEDADR0_PE1: FFC6 54D0 _H | LR8SEDADR1_PE1: FFC6 54D4 _H |
| LR8SEDADR2_PE1: FFC6 54D8 _H | LR8SEDADR3_PE1: FFC6 54DC _H |

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | SEADRmn [17:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SEADRmn[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.35 LRmSEDADRn_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---|---|
| 31 to 18 | Reserved | When read, the value after reset is read. |
| 17 to 0 | SEADRmn[17:0] (m = 1 to 8) (n = 0 to 3) | 1-bit error detection address bank n |

NOTE

Replace the logical address of the lower 4 bits of each bank with the following:

Bank0: SEADRm0[3:0] = 0000_B

Bank1: SEADRm1[3:0] = 0100_B

Bank2: SEADRm2[3:0] = 1000_B

Bank3: SEADRm3[3:0] = 1100_B

(9) LRDEDADRn_PE1 — Local-RAM DED Address Register n (n = 0 to 3)

LRDEDADRn_PE1 (n = 0 to 3) registers are used to hold the address when an error is detected. Error address capture trigger is same as corresponding DEDFn (n = 0 to 3) set in LRDERSTR_PE1. LRDEDADRn_PE1 (n = 0 to 3) are cleared by setting DSTCLRn (n = 0 to 3) in LRDERSTCLR_PE1 register to 1.

Access: LRDEDADRn_PE1 can be read in 32-bit units.

Address: LRDEDADR0_PE1: FFC6 54E0_H
 LRDEDADR1_PE1: FFC6 54E4_H
 LRDEDADR2_PE1: FFC6 54E8_H
 LRDEDADR3_PE1: FFC6 54EC_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | DEADERn [17:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DEADERn[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.36 LRDEDADRn_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------------|---|
| 31 to 18 | Reserved | When read, the value after reset is read. |
| 17 to 0 | DEADRn[17:0] (n = 0 to 3) | 2-bit error detection address bank n |

NOTE

Replace the logical address of the lower 4 bits of each bank with the following:

- Bank0: DEADRm0[3:0] = 0000_B
- Bank1: DEADRm1[3:0] = 0100_B
- Bank2: DEADRm2[3:0] = 1000_B
- Bank3: DEADRm3[3:0] = 1100_B

(10) LRTSTCTL_PE1 — Local RAM Test Control Register

LRTSTCTL_PE1 register controls the ECC error injection. This register must be written while $PROT[1:0] = 01_B$.

Access: LRTSTCTL_PE1 can be read/written in 32-bit units.

Address: FFC6 5604_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|----|----|------------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | ECCTS T | DATSEL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.37 LRTSTCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | ECCTST | ECC Test Mode Bit 0: Normal Mode 1: ECC Test Mode |
| 0 | DATSEL | Data Select Bit ECCTST must be set to 1 0: RAM Data Select 1: ECC bit Select |

CAUTION

When ECC test mode for the local RAM is enabled (ECCTST = 1), the local RAM should be accessed in 4-byte units.

Table 31.38 Table 79.38 Local RAM Test Mode

| ECCTST | DATSEL | Write Data |
|--------|--------|--|
| 0 | 0 | Data and ECC bits are updated (normal mode) |
| 0 | 1 | Data and ECC bits are updated (normal mode) |
| 1 | 0 | Only data is updated. ECC bits are not updated |
| 1 | 1 | Lower 7 bits of data are written as ECC bits. Data is not updated. |

(11) LRDATBF_j_PE1 — Local RAM Test Data Read Buffer j (j = 0, 1)

Access: LRDATBF_j_PE1 can be read in 32-bit units.

Address: LRDATBF0_PE1: FFC6 5608_H
LRDATBF1_PE1: F FC6 560C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | LRDATABF[22:16] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | LRDATABF[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.39 LRDATBF_j_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|---|
| 31 to 23 | Reserved | When read, the value after reset is read. |
| 22 to 16 | LRDATABF [22:16] | ECC Data at Bank (2j + 1) is stored in this register during the read operation from LRAM when ECCTST = 1. |
| 15 to 7 | Reserved | When read, the value after reset is read. |
| 6 to 0 | LRDATABF[6:0] | ECC Data at Bank (2j) is stored in this register during the read operation from LRAM when ECCTST = 1. |

31.2.4.4 Test Function

Through appropriate register setting, desired values can be written to RAM data and the ECC bits. Also, data in the RAM and the ECC bits can all be read.

(1) Writing to RAM data

- (a) Set the ECCTST bit in the local RAM test control register to 1 to enter test mode.
- (b) Set the corresponding DATSEL bit in the local RAM test control register to 0 to select the RAM data to be written to.
- (c) When data is written to the local RAM, only the RAM data is modified without updating the ECC bits.

How to exit this test mode:

- (a) Set the ECCTST bit in the local RAM test control register to 0 to disable test mode (enable normal mode).

(2) Reading RAM data

- (a) Set the ECCDIS bit in the local RAM ECC control register to 1 to disable ECC error detection and correction.
- (b) Read the local RAM. Since neither error detection nor correction is carried out when the local RAM is read, the RAM data is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the local RAM ECC control register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC bits

- (a) Set the ECCTST bit in the local RAM test control register to 1 to enter test mode.
- (b) Set the corresponding DATSEL bit in the local RAM test control register to 1 to select the ECC bits to be written to.
- (c) When data is written to the local RAM, only the ECC bits are modified without updating the RAM data. At that time, bit[6:0] are respectively written to the ECC bits.

How to exit this test mode:

- (a) Set the ECCTST bit in the local RAM test control register to 0 to disable test mode (enable normal mode).

(4) Reading the ECC bits

- (a) Set the ECCTST bit in the local RAM test control register to 1 to enter test mode.
- (b) When the local RAM is read, the ECC bits are stored in the corresponding bank of local RAM test data read buffer 0 or local RAM test data read buffer 1.

How to exit this test mode:

- (a) Set the ECCTST bit in the local RAM test control register to 0 to disable test mode (enable normal mode).

(5) Self-diagnosis of the ECC check function

Desired values can be written to the RAM data and/or ECC bits by following the procedure described in (1) or (3) above. Therefore, a fault can be injected by, for example, inverting appropriate bits of the RAM data and/or ECC bits. After that, self-diagnosis of the ECC decoder is possible by reading the local RAM in normal mode and checking the result of error correction or detection.

31.2.5 Global RAM ECC

31.2.5.1 Overview

The global RAM ECC is summarized in the table below.

Table 31.40 Global RAM ECC

| Item | Description |
|------------------------------------|--|
| ECC error detection and correction | <p>ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit error detection, correction and notification and 2-bit error detection and notification are carried out.</p> |
| Address ECC | <p>Address ECC check can be either enabled or disabled. Address ECC detects address error between Global RAM and master. In the initial state, 1-bit error detection, 2-bit error detection and notification are carried out.</p> |
| Error notification | <p>Upon occurrence of an ECC error or Address ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error.</p> <p>Address ECC Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an address ECC error. <p>In the initial state, error notification is enabled upon detection of an address ECC error.</p> <p>Overflow Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an address buffer overflow error for Data ECC 1-bit error. <p>In the initial state, error notification is enabled upon detection of an address buffer overflow error.</p> <p>ECC 2-bit error, address ECC 2-bit error, ECC 1-bit error, overflow error and address ECC 1-bit error are handled as separate sources. An ECC1-bit error signal is only issued to the ECM when the detected ECC 1-bit error address is different from the address stored in the error address buffer.</p> |
| Error status | <p>A status register is provided, which indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and Address ECC error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.</p> |
| Address capture | <p>Multi-stage address buffers are provided for a 1-bit data error, and check is performed in each stage. The address is latched when the state of each stage is cleared.</p> <ul style="list-style-type: none"> 1-bit data error: 32 stages 2-bit data and address ECC, shared: One stage <p>The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an Address ECC error is detected. The error status also serves as the enable bit of the captured address.</p> |

31.2.5.2 List of Registers

Table 31.41 List of Registers (1/2)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|-------------|---|-----|------------------------|-------------|
| FFC64000 _H | UGRERRINT | Global RAM Error Information Control Register | R/W | 0000 0073 _H | 32/16/8 |
| FFC64004 _H | UGRSERSTCLR | Global-RAM ECC SED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC64008 _H | UGRDERSTCLR | Global RAM ECC DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6400C _H | UGROVFSTR | Global RAM Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC64020 _H | UGRSERSTR | Global RAM ECC SED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC64030 _H | UGRDERSTR | Global RAM ECC DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC64040 _H | UGR1SEDADR | Global RAM 1 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64044 _H | UGR2SEDADR | Global RAM 2 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64048 _H | UGR3SEDADR | Global RAM 3 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6404C _H | UGR4SEDADR | Global RAM 4 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64050 _H | UGR5SEDADR | Global RAM 5 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64054 _H | UGR6SEDADR | Global RAM 6 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64058 _H | UGR7SEDADR | Global RAM 7 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6405C _H | UGR8SEDADR | Global RAM 8 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64060 _H | UGR9SEDADR | Global RAM 9 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64064 _H | UGR10SEDADR | Global RAM 10 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64068 _H | UGR11SEDADR | Global RAM 11 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6406C _H | UGR12SEDADR | Global RAM 12 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64070 _H | UGR13SEDADR | Global RAM 13 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64074 _H | UGR14SEDADR | Global RAM 14 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64078 _H | UGR15SEDADR | Global RAM 15 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6407C _H | UGR16SEDADR | Global RAM 16 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64080 _H | UGR17SEDADR | Global RAM 17 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64084 _H | UGR18SEDADR | Global RAM 18 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64088 _H | UGR19SEDADR | Global RAM 19 ECC SED Address Register | R | 0000 0000 _H | 32 |

Table 31.41 List of Registers (2/2)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|-------------------|--|-----|------------------------|-------------|
| FFC6408C _H | UGR20SEADR | Global RAM 20 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64090 _H | UGR21SEADR | Global RAM 21 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64094 _H | UGR22SEADR | Global RAM 22 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC64098 _H | UGR23SEADR | Global RAM 23 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC6409C _H | UGR24SEADR | Global RAM 24 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640A0 _H | UGR25SEADR | Global RAM 25 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640A4 _H | UGR26SEADR | Global RAM 26 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640A8 _H | UGR27SEADR | Global RAM 27 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640AC _H | UGR28SEADR | Global RAM 28 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640B0 _H | UGR29SEADR | Global RAM 29 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640B4 _H | UGR30SEADR | Global RAM 30 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640B8 _H | UGR31SEADR | Global RAM 31 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640BC _H | UGR32SEADR | Global RAM 32 ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC640C0 _H | UGR00DEADR | Global RAM ECC DED Address Register | R | 0000 0000 _H | 32 |
| FFC64100 _H | GRECCCTL_GRAMC | Global RAM ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC64104 _H | GRTSTCTL | Global RAM Test Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC64108 _H | GRTDATBF0L | Global RAM Test Data Read Buffer (Lower 32 bits of Bank A) | R | 0000 0000 _H | 32/16/8 |
| FFC6410C _H | GRTDATBF0H | Global RAM Test Data Read Buffer (Upper 32 bits of Bank A) | R | 0000 0000 _H | 32/16/8 |
| FFC64110 _H | GRTDATBF1L | Global RAM Test Data Read Buffer (Lower 32 bits of Bank B) | R | 0000 0000 _H | 32/16/8 |
| FFC64114 _H | GRTDATBF1H | Global-RAM Test Data Read Buffer (Upper 32 bits of Bank B) | R | 0000 0000 _H | 32/16/8 |
| FFC64118 _H | GRDECINBF0 | GRAMC ECC Decoder Input Data Buffer 0 | R/W | 0000 0000 _H | 32/16/8 |
| FFC6411C _H | GRDECINBF1 | GRAMC ECC Decoder Input Data Buffer 1 | R/W | 0000 0000 _H | 32/16/8 |
| FFC64200 _H | GRECCCTL_VCI2GRAM | Global RAM ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC64E00 _H | GRECCCTL_AXI2GRAM | Global RAM ECC Control Register | R/W | 0000 0000 _H | 32/16 |

Note: The registers with symbols “_GRAMC”, “_VCI2GRAM” and “_AXI2GRAM” as suffixes are provided for particular ECC controllers corresponding to each access port; the registers with “_GRAMC” are provided for access from the CPU1 to Global RAM and for read-accesses for read-modify-write processing when 8-bit or 16-bit data is written to Global RAM, the registers with “_VCI2GRAM” are provided for access from the system interconnected with Global RAM, and the registers with “_AXI2GRAM” are provided for access from H-bus.

31.2.5.3 Details of Registers

(1) UGRERRINT — Global RAM Error Information Control Register

UGRERRINT register controls whether error information is reported to ECM, when address/data ECC 2-bit error, address/data ECC 1-bit error, and ECC 1-bit error overflow are detected.

Access: UGRERRINT can be read/written in 32/16/8-bit units.

Address: FFC6 4000_H

Value after reset: 0000 0073_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|-------------|------------|--------|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | SEOVFI E | ADEDI E | ASEDIE | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R | R | R/W | R/W |

Table 31.42 UGRERRINT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | SEOVFIE | ECC 1-bit error overflow report enable bit Controls overflow report when 1-bit error overflow flag (SERROVF) in UGROVFSTR register is set. 0: ECC 1-bit error overflow report disabled 1: ECC 1-bit error overflow report enabled |
| 5 | ADEDIE | Address ECC 2-bit error on bus report enable bit Controls error report of 2-bit error detection when address ECC error detection/correction is enabled. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled |
| 4 | ASEDIE | Address ECC 1-bit error on bus report enable bit Controls error report of 1-bit error detection when address ECC error detection/correction is enabled. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | ECC 2-bit error report enable bit Controls error report of 2-bit error detection when ECC error detection/correction is enabled 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled |
| 0 | SEDIE | ECC 1-bit error report enable bit Controls error report of 1-bit error detection when ECC error detection/correction is enabled 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled |

(2) UGRSERSTCLR — Global RAM ECC SED Status Clear Register

UGRSERSTCLR register is used to clear SEDF[31:0] in UGRSERSTR, SERROVF in UGROVFSTR, and error address in UGRnSEDADR (n = 1 to 32). This is write only register and read value is always “0”. UGRSERSTCLR register has a lower priority than a set factor. Priority is given to a set factor when UGRSERSTCLR and a set factor compete. A set factor is a trigger for setting SEDF[31:0] in UGRSERSTR, setting SERROVF in UGROVFSTR, or capturing an error address in UGRnSEDADR (n = 1 to 32).

Access: UGRSERSTCLR can be written in 32/16/8-bit units.

Address: FFC6 4004_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | SSTCLR[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SSTCLR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 31.43 UGRSERSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 0 | SSTCLR[31:0] | Data ECC 1-bit error flag clear n-th 1: Following flag and address clear UGRSERSTR.SEDF(n-1), UGROVFSTR.SERROVF, UGRnSEDADR.SEADR |

(3) UGRDERSTCLR — Global RAM ECC DED Status Clear Register

UGRDERSTCLR register is used to clear ADEDF/ASEDF and DEDF in UGRDERSTR and error address in UGR00DEDADR. This is write only register and read value is always “0”.

UGRDERSTCLR register has a lower priority than a set factor. Priority is given to a set factor when UGRDERSTCLR and a set factor compete. A set factor is a trigger for setting ADEDF/ASEDF and DEDF in UGRDERSTR.

Access: UGRDERSTCLR can be written in 32/16/8-bit units.

Address: FFC6 4008_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.44 UGRDERSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | DSTCLR | Data ECC 2-bit error flag and address ECC error flag clear 1: Following flags and addresses are cleared: UGRDERSTR.ADEDF, UGRDERSTR.ASEDF, UGRDERSTR.DEDF, UGR00DEDADR.DEADR |

(4) UGROVFSTR — Global RAM Error Count Overflow Status Register

UGROVFSTR register monitors occurrence of error overflow. Overflow occurs when an SED for which the error address has not been captured is detected while ECC 1-bit error status register is full. If the ECC 1-bit error status register is full and the error address is the same as one of the error addresses already captured, this flag is not set. SERROVF flag is cleared when at least one bit of SSTCLR[31:0] in UGRSERSTCLR register is asserted.

Access: UGROVFSTR can be read in 32/16/8-bit units.

Address: FFC6 400C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.45 UGROVFSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | SERROVF | 1-bit error overflow flag When all bits of the ECC 1-bit error status register (UGRSERSTR.SEDF[31:0]) are "1", an ECC 1 bit error occurs and the error address does not match any of the captured error addresses, this flag is set. |

(5) UGRSERSTR — Global RAM ECC SED Status Register

UGRSERSTR is the error monitor register. Each error flag is “0” and when a new error occurs, an error status flag is set. An error flag is set at the lowest number empty bit of UGRSERSTR (e.g. If SEDF[0][1][3] have been set to “1” and all other bits have been empty, the next flag is set at SEDF[2]). If multiple SEDs causes are simultaneously detected and there are sufficient empty bits, all detected SEDs are set (e.g. if a SED and another SED which occurs at a different address are detected simultaneously, both SEDFs are set.). However, if SEDs occur at a same address simultaneously, only one of errors is set according to the fixed priority. The priority order is PE1, system interconnect and H-bus. (e.g. if a SED which is input from PE1 path and an SED which is input from another path are detected simultaneously and those occur at the same address, only SEDF from the PE1 path is set.) Furthermore, an error address which has already been captured in UGRnSEDAADR (n = 1 to 32) must not be captured again. UGRSERSTR register is cleared by setting SSTCLR[31:0] in UGRSERSTCLR register to 1.

Access: UGRSERSTR can be read in 32/16/8-bit units.

Address: FFC6 4020_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | SEDF[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SEDF[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.46 UGRSERSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 0 | SEDF[31:0] | ECC 1-bit error monitor flag SEDF(n-1) in n-th buffer Condition to “0”: write “1” to SSTCLR(n-1) in UGRSERSTCLR Condition to “1”: ECC 1-bit error is detected. |

(6) UGRDERSTR — Global RAM ECC DED Status Register

UGRDERSTR is the error monitor register. All error flags are “0” and when a new error occurs, an error status flag is set. If multiple error causes which are input from different access ports are detected simultaneously, only one of detected errors is set according to the fixed priority. The reason is that there is only one 2-bit error Address Register. The priority order is PE1, system interconnect and H-bus. (e.g. if a DED which is input from PE1 path and an ASED which is input from another path are detected simultaneously and those occur at the same address, only DEDF from the PE1 path is set.) If multiple error causes occur from same access port simultaneously, all detected errors are set. (e.g. if DED and ASED are simultaneously input from PE1 path, both DEDF and ASEDF are set.) UGRDERSTR register is cleared by setting DSTCLR in UGRDERSTR register to 1.

Access: UGRDERSTR can be read in 32/16/8-bit units.

Address: FFC6 4030_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|-------|-------|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | ADEDF | ASEDF | — | — | — | DEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.47 UGRDERSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 6 | Reserved | When read, the value after reset is read. |
| 5 | ADEDF | Address ECC 2-bit error monitor flag Condition to “0”: write “1” to DSTCLR in UGRDERSTCLR Condition to “1”: ADEDF is “0” and address ECC 2-bit error is detected. |
| 4 | ASEDF | Address ECC 1-bit error monitor flag Condition to “0”: write “1” to DSTCLR in UGRDERSTCLR Condition to “1”: ASEDF is “0” and address ECC 1-bit error is detected. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | DEDF | ECC 2-bit error monitor flag Condition to “0”: write “1” to DSTCLR in UGRDERCTCLR Condition to “1”: DEDF is “0” and ECC 2-bit error is detected. |

Note: This register has been named for capturing ECC 2-bit error (DED) error status, but the register captures Address ECC 1-bit error (ASED) too. Because ASED is also an emergency error and S/W should check the status immediately, ASED is treated like a DED error.

(7) UGRnSEDADR — Global RAM n ECC SED Address Register (n = 1 to 32)

UGRnSEDADR registers are used to hold the address when a 1-bit error is detected. Error address capture trigger is same as corresponding SEDF (n – 1) set in UGRSERSTR. UGRnSEDADR are cleared by setting SSTCLR (n – 1) in UGRSERSTCLR register to 1.

Access: UGRnSEDADR can be read in 32-bit units.

Address: UGR1SEDADR: FFC6 4040_H
 UGR2SEDADR: FFC6 4044_H
 UGR3SEDADR: FFC6 4048_H
 UGR4SEDADR: FFC6 404C_H
 UGR5SEDADR: FFC6 4050_H
 UGR6SEDADR: FFC6 4054_H
 UGR7SEDADR: FFC6 4058_H
 UGR8SEDADR: FFC6 405C_H
 UGR9SEDADR: FFC6 4060_H
 UGR10SEDADR: FFC6 4064_H
 UGR11SEDADR: FFC6 4068_H
 UGR12SEDADR: FFC6 406C_H
 UGR13SEDADR: FFC6 4070_H
 UGR14SEDADR: FFC6 4074_H
 UGR15SEDADR: FFC6 4078_H
 UGR16SEDADR: FFC6 407C_H
 UGR17SEDADR: FFC6 4080_H
 UGR18SEDADR: FFC6 4084_H
 UGR19SEDADR: FFC6 4088_H
 UGR20SEDADR: FFC6 408C_H
 UGR21SEDADR: FFC6 4090_H
 UGR22SEDADR: FFC6 4094_H
 UGR23SEDADR: FFC6 4098_H
 UGR24SEDADR: FFC6 409C_H
 UGR25SEDADR: FFC6 40A0_H
 UGR26SEDADR: FFC6 40A4_H
 UGR27SEDADR: FFC6 40A8_H
 UGR28SEDADR: FFC6 40AC_H
 UGR29SEDADR: FFC6 40B0_H
 UGR30SEDADR: FFC6 40B4_H
 UGR31SEDADR: FFC6 40B8_H
 UGR32SEDADR: FFC6 40BC_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|----|--------------|----|----|----|----|
| | — | | | | | | | | | | | SEADR[20:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SEADR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.48 UGRnSEDADR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. |
| 20 to 0 | SEADR[20:0] | 1-bit error detection address |

(8) UGR00DEDADR — Global RAM ECC DED Address Register

UGR00DEDADR register is used to hold the address when an error is detected. Error address capture trigger is same as corresponding ADEDF/ASEDF/DEDF set in UGRDERSTR. If this register has captured an error address, it does not capture any more address. This register stores an address without correction even if the error is 1-bit error. UGR00DEDADR is cleared by setting DSTCLR in UGRDERSTCLR register to 1.

Access: UGR00DEDADR can be read in 32-bit units.

Address: FFC6 40C0_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|--------------|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | DEADR[20:16] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DEADR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.49 UGR00DEDADR Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 21 | Reserved | When read, the value after reset is read. |
| 20 to 0 | DEADR[20:0] | Data 2-bit error or address 1 bit or address 2-bit error detection address |

(9) GRECCCTL_GRAMC — Global RAM ECC Control Register

GRECCCTL_GRAMC register controls the ECC error detection/correction and 1-bit error correction on address and data ECC. Writing ECC control registers must be executed with PROT[1:0] = 01_B.

Access: GRECCCTL_GRAMC can be read/written in 32/16-bit units.

Address: FFC6 4100_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|---------|---------|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | ASECDIS | AECDDIS | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R/W | R/W |

Table 31.50 GRECCCTL_GRAMC Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enabled Other: write is disabled These bits are always read as 0. |
| 13 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | ASECDIS | Address ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Address ECC error detection/correction (AECDDIS = 0). 0: Correction is carried out when 1-bit error is detected 1: Correction is not carried out when 1-bit error is detected |
| 3 | AECDDIS | Address ECC disable bit Enables or disables ECC error detection/correction. 0: ECC error detection/correction is enable 1: ECC error detection/correction is disable |
| 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | Data ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using ECC error detection/correction (ECCDIS = 0). enable/disable. 0: Correction is carried out when 1-bit error is detected 1: Correction is not carried out when 1-bit error is detected |
| 0 | ECCDIS | Data ECC disable bit Enables or disables ECC error detection/correction. 0: ECC error detection/correction is enable 1: ECC error detection/correction is disable |

(10) GRTSTCTL — Global RAM Test Control Register

GRTSTCTL register controls the ECC error injection. This register must be written while PROT[1:0] = 01_B.

Access: GRTSTCTL can be read/written in 32/16-bit units.

Address: FFC6 4104_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|------------|-------------|-------------|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | ECCTS T | DECIN EN | DATSEL[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |

Table 31.51 GRTSTCTL Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | ECCTST | ECC Test Mode Bit 0: Normal Mode 1: ECC Test Mode |
| 2 | DECINEN | GRAMC ECC Decoder Error Injection Enable Bit ECCTST must be set to 1. 0: ECC decoder input buffer data is not selected 1: ECC decoder input buffer data is selected |

Table 31.51 GRTSTCTL Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|-------------|--|
| 1, 0 | DATSEL[1:0] | <p>Read Buffer Storage Data Select 0 and 1</p> <p>These bits are valid when ECCTST = 1. These bits select the value to be stored in the read buffer GRTDATBFn, the value to be written to each field. Set (PROT1, PROT0) = (0, 1) simultaneously when writing to this bit.</p> <p>00:</p> <ul style="list-style-type: none"> – GRTDATBFn: When a read access involves an RmW, the ECC bits are stored. – Global RAM: When a write access involves an RmW, the data area that is updated depends on the size and the address of write access. The ECC bits are not updated. <p>01:</p> <ul style="list-style-type: none"> – GRTDATBFn: When a read access involves an RmW, the ECC bits are stored. – Global RAM: When a write access involves an RmW, only the ECC bits are updated. The data area is not updated. <p>10:</p> <ul style="list-style-type: none"> – GRTDATBFn: When access is RmW, this register holds the result of ECC decoding for the data read out in the read portion of the RmW cycle. Its value is not updated in the case of access that is not RmW. – Global RAM: Operates in the same way as in normal operating mode (i.e. when ECCTST = 0). <p>11:</p> <ul style="list-style-type: none"> – GRTDATBFn: When access is RmW, this register holds the result of ECC decoding for use in the updating of data at the time of RmW access. Its value is not updated in the case of access that is not RmW. – Global RAM: Operates in the same way as in normal operating mode (i.e. when ECCTST = 0). <p>In any case, the result of reading by the CPU, DMAC, etc. is the same as the value that would normally be read out.</p> |

(11) GRTDATBF[0/1][L/H] — Global RAM Test Data Read Buffer (Lower/Upper 32 bits of BankA(0)/B(1))

Access: GRTDATBF can be read in 32/16/8-bit units.

Address: GRTDATBF0L: FFC6 4108_H
 GRTDATBF0H: FFC6 410C_H
 GRTDATBF1L: FFC6 4110_H
 GRTDATBF1H: FFC6 4114_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| GRTDATBF[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GRTDATBF[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.52 GRTDATBF Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|--|
| 31 to 0 | GRTDATBF [31:0] | <p>These bits are valid when ECCTST = 1 (selecting test mode) in the global RAM test control register.</p> <ul style="list-style-type: none"> When (DATSEL1, DATSEL0) = (0, 0) or (0, 1) When reading from the RAM, the ECC bits are respectively stored in GRTDATBF[6:0]. 0 is stored in GRTDATBF[31:7]. When (DATSEL1, DATSEL0) = (1, 0) When access is RmW, the output data from the ECC decoder at the time of reading (after updating) is stored in GRTDATBF[31:0]. When (DATSEL1, DATSEL0) = (1, 1) When access is RmW, the output data from the ECC decoder which is used for update (after update) is stored in GRTDATBF[31:0]. |

Note: GRTDATBF0L : For Global RAM Bank A lower 32 bits
 GRTDATBF0H : For Global RAM Bank A upper 32 bits
 GRTDATBF1L : For Global RAM Bank B lower 32 bits
 GRTDATBF1H : For Global RAM Bank B upper 32 bits

(12) GRDECINBF0 — GRAMC ECC Decoder Input Data Buffer 0

Access: GRDECINBF0 can be read/written in 32/16/8-bit units.

Address: FFC6 4118_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | GRDECINBF0[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | GRDECINBF0[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.53 GRDECINBF0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 31 to 0 | GRDECINBF0 [31:0] | <p>These bits are valid when ECCTST = 1 (selecting test mode) in the global RAM test control register.</p> <p>When DECINEN = 1, the value of this register is input to the ECC decoder as data for use in updating in response to the execution of an RmW instruction. The value is treated as if it were 32 bits of data from RAM.</p> <p>This register is common to banks A and B and provides the values for both the 32 higher-order bits and the 32 lower-order bits.</p> |

(13) GRDECINBF1 — GRAMC ECC Decoder Input Data Buffer 1

Access: GRDECINBF1 can be read/written in 32/16/8-bit units.

Address: FFC6 411C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|-----------------|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | GRDECINBF1[6:0] | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.54 GRDECINBF1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | GRDECINBF1 [6:0] | These bits are valid when ECCTST = 1 (selecting test mode) in the global RAM test control register. When DECINEN = 1, the value of this register is input to the ECC decoder as data for use in updating in response to the execution of an RmW instruction. The value is treated as if it were 7 bits of data from the ECC. This register is common to banks A and B and provides the values for both the 32 higher-order bits and the 32 lower-order bits. |

(14) GRECCCTL_VCI2GRAM/AXI2GRAM — Global RAM ECC Control Register

GRECCCTL_VCI2GRAM/AXI2GRAM registers control the ECC error detection/correction and 1-bit error correction. Writing ECC control registers must be executed with PROT[1:0] = 01_B.

Access: GRECCCTL_VCI2GRAM/AXI2GRAM can be read/written in 32/16-bit units.

Address: GRECCCTL_VCI2GRAM: FFC6 4200_H
GRECCCTL_AXI2GRAM: FFC6 4E00_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.55 GRECCCTL_VCI2GRAM/AXI2GRAM Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | Data ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using ECC error detection/correction (ECCDIS = 0). 0: Correction is carried out when 1-bit is detected 1: Correction is not carried out when 1-bit is detected |
| 0 | ECCDIS | Data ECC disable bit Enables or disables ECC error detection/correction. 0: ECC error detection/correction is enable 1: ECC error detection/correction is disable |

31.2.5.4 Test Function

Through appropriate register setting, desired values can be written to RAM data and the ECC bits. Also, data in the RAM and the ECC bits and the ECC decoder output data for RmW can all be read. It is possible to input the desired data in the ECC decoder for RmW. When using the test function, set the WTBufMode bit in the WTbuf configuration register to "0" to disable the write-through buffer.

(1) Writing to RAM data

- (a) Set the ECCTST bit in the global RAM test control register to 1 to enter test mode.
- (b) Set the corresponding DATSEL1 to 0 and DATSEL0 to 0 in the global RAM test control register to select the RAM data to be written to.
- (c) When data is written to the global RAM, only the RAM data is modified without updating the ECC bits.

How to exit test this mode:

- (a) Set the ECCTST bit in the global RAM test control register to 0 to disable test mode (enable normal mode).

(2) Reading RAM data

- (a) Set the ECCDIS bit in the global RAM ECC control register to 1 to disable ECC error detection and correction.
- (b) Read the global RAM. Since neither error detection nor correction is carried out when the global RAM is read, the RAM data is read unchanged.

How to exit this test mode:

- (a) Set the ECCDIS bit in the global RAM ECC control register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC bits

- (a) Set the ECCTST bit in the global RAM test control register to 1 to enter test mode.
- (b) Set the corresponding DATSEL1 bit to 0 and DATSEL0 bit to 1 in the global RAM test control register to select the ECC bits to be written to.
- (c) When data is written to the global RAM, only the ECC bits are modified without updating the RAM data. At that time, bit[6:0] are respectively written to the ECC bits.

How to exit this test mode:

- (a) Set the ECCTST bit in the global RAM test control register to 0 to disable test mode (enable normal mode).

(4) Reading the ECC bits

- (a) Set the ECCTST bit in the global RAM test control register to 1 to enter test mode.
- (b) Set the corresponding DATSEL1 bit to 0 and DATSEL0 bit to 1 in the global RAM test control register to select the ECC bits to be read.
- (c) When data in the global RAM is read, the ECC bits are stored in the corresponding Global RAM test data read buffer registers.

How to exit this test mode:

- (a) Set the ECCTST bit in the global RAM test control register to 0 to disable test mode (enable normal mode).

(5) Self-diagnosis of the ECC check function for the access ports

Desired values can be written to the RAM data and/or ECC bits by following the procedure described in (1) or (3) above. Therefore, a fault can be injected by, for example, inverting appropriate bits of the RAM data and/or ECC bits. After that, self-diagnosis of the ECC decoder is possible by reading the global RAM in normal mode and checking the result of error correction or detection.

(6) Self-diagnosis of the ECC decoder for the data read out in an RmW operation

- (a) Inject suitable erroneous values as RAM data or to the ECC bits by following procedure (1) or (3) above.
- (b) Set the DATSEL1 bit to 1 and the DATSEL0 bit to 0 in the global RAM test control register to make the output data from the ECC decoder for the data read out in an RmW operation the target for reading.
- (c) After RmW processing for the global RAM proceeds, the data read out in an RmW operation for global RAM are stored in the corresponding Global RAM test data read buffer registers. Checking the result allows self-diagnosis of the ECC decoder for the data read out in an RmW operation.

(7) Self-diagnosis of the ECC decoder for the data updated during RmW access

- (a) Set the DATSEL1 and DATSEL0 bits to 1 in the global RAM test control register to make the output data from the ECC decoder for the data updated during an RmW operation the target for reading.
- (b) Through the above setting, the data input from the ECC decoder for the data updated during RmW access is switched to data from the ECC decoder input buffer 0 or 1 (GRDECINBF0 or 1) instead of write data sent from the access ports. As a result, suitable erroneous values can be injected by setting an appropriate value in ECC decoder input buffer 0 or 1.

31.2.6 Instruction Cache ECC and EDC

31.2.6.1 Overview

The instruction cache ECC is summarized in the table below.

Table 31.56 Instruction Cache ECC

| Item | Description |
|---------------------|--|
| ECC error detection | <p>ECC error detection can be either enabled or disabled.</p> <p>When enabled, the following setting can be selected.</p> <ul style="list-style-type: none"> ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). <p>When disabled, error detection is not carried out.</p> <p>In the initial state, the ECC function is enabled, and 1-bit error detection, and notification, and 2-bit error detection and notification are carried out by the instruction cache.</p> |
| Error notification | <p>Upon occurrence of an ECC error, it is notified to the Error Control Module.</p> <p>ECC Error:</p> <ul style="list-style-type: none"> Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. <p>In the initial state, notification of the ECC 2-bit error is enabled, and notification of the ECC 1-bit error is enabled.</p> <p>An ECC1-bit error signal is only issued to the ECM when the detected ECC 1-bit error address is different from the address stored in the error address buffer.</p> |
| Error status | <p>A status register is provided, which indicates the status of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set.</p> <p>The error status can be cleared using the clear register.</p> |
| Address capture | <p>If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured.</p> <p>The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected.</p> <p>The error status also serves as the enable bit of the captured address.</p> |

31.2.6.2 List of Registers

Table 31.57 List of Registers

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|----------------|---|-----|------------------------|-------------|
| FFC60400 _H | IDECCCTL_PE1 | Instruction Cache Data RAM Data ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC60404 _H | IDERRINT_PE1 | Instruction Cache Data RAM Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC60408 _H | IDSERSTCLR_PE1 | Instruction Cache Data RAM Data ECC SED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6040C _H | IDDERSTCLR_PE1 | Instruction Cache Data RAM Data ECC DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC60410 _H | IDOVFSTR_PE1 | Instruction Cache Data RAM Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC60420 _H | IDSERSTR_PE1 | Instruction Cache Data RAM Data ECC SED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC60450 _H | IDDERSTR_PE1 | Instruction Cache Data RAM Data ECC DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC60460 _H | IDSEDADR0_PE1 | Instruction Cache Data RAM(Bank0) Data ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC60464 _H | IDSEDADR1_PE1 | Instruction Cache Data RAM(Bank1) Data ECC SED Address Register | R | 0000 0000 _H | 32 |
| FFC604E0 _H | IDDEDADR0_PE1 | Instruction Cache Data RAM(Bank0) Data ECC DED Address Register | R | 0000 0000 _H | 32 |
| FFC604E4 _H | IDDEDADR1_PE1 | Instruction Cache Data RAM(Bank1) Data ECC DED Address Register | R | 0000 0000 _H | 32 |
| FFC61400 _H | ITECCCTL_PE1 | Instruction Cache Tag RAM ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC61404 _H | ITERRINT_PE1 | Instruction Cache Tag RAM Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC61408 _H | ITSERSTCLR_PE1 | Instruction Cache Tag RAM ECC SED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6140C _H | ITDERSTCLR_PE1 | Instruction Cache Tag RAM ECC DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC61410 _H | ITOVFSTR_PE1 | Instruction Cache Tag RAM Error Count Overflow Status Register | R | xxxx xxxx _H | 32/16/8 |
| FFC61420 _H | ITSERSTR_PE1 | Instruction Cache Tag RAM ECC SED Status Register | R | 0000 0000 _H | 32 |
| FFC61450 _H | ITDERSTR_PE1 | Instruction Cache Tag RAM ECC DED Status Register | R | 0000 0000 _H | 32 |
| FFC61460 _H | ITSEDADR_PE1 | Instruction Cache Tag RAM SED Address Register | R | 0000 0000 _H | 32 |
| FFC614E0 _H | ITDEDADR_PE1 | Instruction Cache Tag RAM DED Address Register | R | 0000 0000 _H | 32 |

31.2.6.3 Details of Registers

(1) IDECCCTL_PE1 — Instruction Cache Data RAM ECC Control Register

IDECCCTL_PE1 register controls the ECC error detection. Writing ECC control registers can only be executed with PROT[1:0] = 01_B. ECC error correction is controlled by D1EIV bit of ICCTRL register (refer to **Section 3, CPU System**).

Access: IDECCCTL_PE1 can be read/written in 32/16-bit units.

Address: FFC6 0400_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 31.58 IDECCCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | ECCDIS | ECC disable bit Enables or disables ECC error detection. 0: ECC error detection is enabled 1: ECC error detection is disabled |

(2) IDERRINT_PE1 — Instruction Cache Data RAM Error Information Control Register

IDERRINT_PE1 register controls whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected.

Access: IDERRINT_PE1 can be read/written in 32/16/8-bit units.

Address: FFC6 0404_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.59 IDERRINT_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | ECC 2-bit error notification enable bit Controls error notification of 2-bit error detection when ECC error detection is enabled 0: ECC 2-bit error notification disabled 1: ECC 2-bit error notification enabled |
| 0 | SEDIE | ECC 1-bit error notification enable bit Controls error notification of 1-bit error detection when ECC error detection is enabled 0: ECC 1-bit error notification disabled 1: ECC 1-bit error notification enabled |

(3) IDSERSTCLR_PE1 — Instruction Cache Data RAM Data ECC SED Status Clear Register

IDSERSTCLR_PE1 register is used to clear SEDF0/1 in IDSERSTR_PE1 and SERROVF0/1 in IDOVFSTR_PE1, and error address in IDSEDADR0/1_PE1. This is write only register and read value is always “0”.

Access: IDSERSTCLR_PE1 can be written in 32/16/8-bit units.

Address: FFC6 0408_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR1 | SSTCLR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W | W |

Table 31.60 IDSERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When writing, write the value after reset. |
| 1 | SSTCLR1 | 1-bit error flags clear (bank 1) 1: All of the following 1-bit error flags are cleared. IDSERSTR_PE1.SEDF1, IDOVFSTR_PE1.SERROVF1, IDSEDADR1_PE1.SEADR |
| 0 | SSTCLR0 | 1-bit error flags clear (bank 0) 1: All of the following 1-bit error flags are cleared. IDSERSTR_PE1.SEDF0, IDOVFSTR_PE1.SERROVF0, IDSEDADR0_PE1.SEADR |

(4) IDDERSTCLR_PE1 — Instruction Cache Data RAM Data ECC DED Status Clear Register

IDDERSTCLR_PE1 register is used to clear DEDF0/1 in IDDERSTR_PE1 and error address in IDDEDADR0/1_PE1. This is write only register and read value is always “0”.

Access: IDDERSTCLR_PE1 can be written in 32/16/8-bit units.

Address: FFC6 040C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DSTCLR1 | DSTCLR0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W | W |

Table 31.61 IDDERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When writing, write the value after reset. |
| 1 | DSTCLR1 | 2-bit error flags clear (bank 1) 1: All 2-bit error flags clear IDDERSTR_PE1.DEDF1, IDDEDADR1_PE1.DEADR |
| 0 | DSTCLR0 | 2-bit error flags clear (bank 0) 1: All 2-bit error flags clear IDDERSTR_PE1.DEDF0, IDDEDADR0_PE1.DEADR |

(5) IDOVFSTR_PE1 — Instruction Cache Data RAM Error Count Overflow Status Register

IDOVFSTR_PE1 register monitors occurrence of error overflow. ECC 1bit error occurs at same error address is detected while the error status is full. If the error status is full and the same error (same error address) has occurred, then this flag is not set. SERROVF0/1 flag is cleared by setting SSTCLR0/1 in IDSERSTCLR_PE1 register to 1.

Access: IDOVFSTR_PE1 can be read in 32/16/8-bit units.

Address: FFC6 0410_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SERROVF1 | SERROVF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.62 IDOVFSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | SERROVF1 | 1-bit error overflow flag (bank1) When SEDF1 in IDSERSTR_PE1 is "1", an ECC 1bit error occurs and the error address does not match any of the captured error addresses, this flag is set. |
| 0 | SERROVF0 | 1-bit error overflow flag (bank0) When SEDF0 in IDSERSTR_PE1 is "1", an ECC 1bit error occurs and the error address does not match any of the captured error addresses, this flag is set. |

(6) IDSERSTR_PE1 — Instruction Cache Data RAM Data ECC SED Status Register

IDSERSTR_PE1 is the error monitor register. IDSERSTR_PE1 monitors occurrence of the first error. Error detection and flag update condition are individually set in bank 0 and 1. IDSERSTR_PE1 register is cleared by setting SSTCLR1/0 in IDSERSTCLR_PE1 register to 1.

Access: IDSERSTR_PE1 can be read in 32/16/8-bit units.

Address: FFC6 0420_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | SEDF1 | — | — | — | — | — | — | — | SEDF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.63 IDSERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 9 | Reserved | When read, the value after reset is read. |
| 8 | SEDF1 | ECC 1-bit error monitor flag (bank 1) Condition for "0": write "1" to SSTCLR1 in IDSERSTCLR_PE1 Condition for "1": SEDF1 is "0" and ECC 1-bit error is detected. |
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | SEDF0 | ECC 1-bit error monitor flag (bank 0) Condition for "0": write "1" to SSTCLR0 in IDSERSTCLR_PE1 Condition for "1": SEDF0 is "0" and ECC 1-bit error is detected. |

(7) IDDERSTR_PE1 — Instruction Cache Data RAM Data ECC DED Status Register

IDDERSTR_PE1 is the error monitor register. IDDERSTR_PE1 monitors occurrence of the first error. Error detection and flag update condition are individually set in bank 0 and 1. IDDERSTR_PE1 register is cleared by setting DSTCLR1/0 in IDDERSTCLR_PE1 register to 1.

Access: IDDERSTR_PE1 can be read in 32/16/8-bit units.

Address: FFC6 0450_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | DEDF1 | — | — | — | — | — | — | — | DEDF0 | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.64 IDDERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 10 | Reserved | When read, the value after reset is read. |
| 9 | DEDF1 | ECC 2-bit error monitor flag (bank 1) Condition to "0": write "1" to DSTCLR1 in IDDERSTCLR_PE1 Condition to "1": DEDF1 is "0" and ECC 2-bit error is detected. |
| 8 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF0 | ECC 2-bit error monitor flag (bank 0) Condition to "0": write "1" to DSTCLR0 in IDDERSTCLR_PE1 Condition to "1": DEDF0 is "0" and ECC 2-bit error is detected. |
| 0 | Reserved | When read, the value after reset is read. |

(8) IDSEDADR0/1_PE1 — Instruction Cache Data RAM (Bank0/1) Data ECC SED Address Register

IDSEDADR0/1_PE1 registers are used to hold the address when an SED error is detected. Error address capture trigger is same as corresponding SEDF0/1 set in IDSERSTR_PE1. Since this register holds the internal address, add the base address of the associated memory to transform the internal address to the actual address. IDSEDADR0/1_PE1 are cleared by setting SSTCLR1/0 in IDSERSTCLR_PE1 register to 1.

Access: IDSEDADR0/1_PE1 can be read in 32-bit units.

Address: IDSEDADR0_PE1: FFC6 0460_H
IDSEDADR1_PE1: FFC6 0464_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | SEADR[8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.65 IDSEDADR0/1_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 9 | Reserved | When read, the value after reset is read. |
| 8 to 0 | SEADR[8:0] | 1-bit error detection address bank n (n = 0 or 1) |

(9) IDDEDADR0/1_PE1 — Instruction Cache Data RAM (Bank0/1) Data ECC DED Address Register

IDDEDADR0/1_PE1 registers are used to hold the address when a DED error is detected. Error address capture trigger is same as corresponding DEDF0/1 set in IDDERSTR_PE1. Since this register holds the internal address, add the base address of the associated memory to transform the internal address to the actual address. IDDEDADR0/1_PE1 are cleared by setting DSTCLR0/1 in IDDERSTCLR_PE1 register to 1.

Access: IDDEDADR0/1_PE1 can be read in 32-bit units.

Address: IDDEDADR0_PE1: FFC6 04E0_H
IDDEDADR1_PE1: FFC6 04E4_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | DEADR[8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.66 IDDEDADR0/1_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 9 | Reserved | When read, the value after reset is read. |
| 8 to 0 | DEADR[8:0] | 2-bit error detection address bank n (n = 0 or 1) |

(10) ITECCCTL_PE1 — Instruction Cache Tag RAM ECC Control Register

ITECCCTL_PE1 register controls the ECC error detection. Writing ECC control registers can only be executed with PROT[1:0] = 01_B.

Access: ITECCCTL_PE1 can be read/written in 32/16-bit units.

Address: FFC6 1400_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 31.67 ITECCCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | ECCDIS | ECC disable bit Enables or disables ECC error detection. 0: ECC error detection is enabled 1: ECC error detection is disabled |

(11) ITERRINT_PE1 — Instruction Cache Tag RAM Error Information Control Register

ITERRINT_PE1 register controls whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected.

Access: ITERRINT_PE1 can be read/written in 32/16/8-bit units.

Address: FFC6 1404_H

Value after reset: 0000 0003_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.68 ITERRINT_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | ECC 2-bit error report enable bit Controls error report of 2-bit error detection when ECC error detection is enabled 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled |
| 0 | SEDIE | ECC 1-bit error report enable bit Controls error report of 1-bit error detection when ECC error detection is enabled 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled |

(12) ITSERSTCLR_PE1 — Instruction Cache Tag RAM ECC SED Status Clear Register

ITSERSTCLR_PE1 register is used to clear SEDF in ITSERSTR_PE1 and SERROVF in ITOVFSTR_PE1, and error address in ITSEDADR_PE1. This is write only register and read value is always “0”.

Access: ITSERSTCLR_PE1 can be written in 32/16/8-bit units.

Address: FFC6 1408_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.69 ITSERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | 1-bit error flag clear 0: Nothing operates 1: All 1-bit error flags clear ITSERSTR_PE1.SEDF, ITOVFSTR_PE1.SERROVF, ITSEDADR_PE1.SEADR |

(13) ITDERSTCLR_PE1 — Instruction Cache Tag RAM ECC DED Status Clear Register

ITDERSTCLR_PE1 register is used to clear DEDF in ITDERSTR_PE1, and error address in ITDEDADR_PE1. This is write only register and read value is always “0”.

Access: ITDERSTCLR_PE1 can be written in 32/16/8-bit units.

Address: FFC6 140C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.70 ITDERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | DSTCLR | 2-bit error flag clear 0: Nothing operates 1: All 2-bit error flags clear ITDERSTR_PE1.DEDF, ITDEDADR_PE1.DEADR |

(14) ITOVFSTR_PE1 — Instruction Cache Tag RAM Error Count Overflow Status Register

ITOVFSTR_PE1 register monitors occurrence of error overflow. Overflow occurs when different overflow*¹ is detected while the error status is full. If the error status is full and the same error (same error cause and same error address) has occurred, then this flag is not set. SERROVF flag is cleared by setting SSTCLR in ITSERSTCLR_PE1 register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: ITOVFSTR_PE1 can be read in 32/16/8-bit units.

Address: FFC6 1410_H

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.71 ITOVFSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 5 | Reserved | When read, the value after reset is read. |
| 4 | Reserved | This bit is undefined. |
| 3 to 1 | Reserved | When read, the value after reset is read. |
| 0 | SERROVF | 1-bit error overflow flag When SEDF in ITSERSTR_PE1 is "1", an ECC 1bit error occurs and the error address does not match any of the captured error addresses, this flag is set. |

(15) ITSERSTR_PE1 — Instruction Cache Tag RAM ECC SED Status Register

ITSERSTR_PE1 is the error monitor register for SED. If the error flag is “0” for a new error occurs, error status flag is set. ITSERSTR_PE1 register is cleared by setting SSTCLR in ITSERSTCLR_PE1 register to 1.

Access: ITSERSTR_PE1 can be read only in 32-bit units.

Address: FFC6 1420_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.72 ITSERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | SEDF | ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in ITSERSTCLR_PE1 Condition to “1”: SEDF is “0” and ECC 1-bit error is detected. |

(16) ITDERSTR_PE1 — Instruction Cache Tag RAM ECC DED Status Register

ITDERSTR_PE1 is the DED error monitor register. If the error flag is “0” and a new error occurs, error status flag is set. ITDERSTR_PE1 register is cleared by setting DSTCLR in ITDERSTCLR_PE1 register to 1.

Access: ITDERSTR_PE1 can be read in 32-bit units.

Address: FFC6 1450_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDF | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.73 ITDERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF | ECC 2-bit error monitor flag Condition to “0”: write “1” to DSTCLR in ITDERSTCLR_PE1 Condition to “1”: DEDF is “0” and ECC 2-bit error is detected. |
| 0 | Reserved | When read, the value after reset is read. |

(17) ITSEDADR_PE1 — Instruction Cache Tag RAM SED Address Register

ITSEDADR_PE1 register is used to hold the address when an SED error is detected. Error address capture trigger is same as corresponding SEDF set in ITSERSTR_PE1. Entry address is stored in SEADR[7:0] and way group number in SEADR[8] in this register. ITSEDADR_PE1 is cleared by setting SSTCLR in ITSERSTCLR_PE1 register to 1.

Access: ITSEDADR_PE1 can be read in 32-bit units.

Address: FFC6 1460_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | SEADR[8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.74 ITSEDADR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 9 | Reserved | When read, the value after reset is read. |
| 8 to 0 | SEADR[8:0] | 1-bit error detection address |

(18) ITDEDADR_PE1 — Instruction Cache Tag RAM DED Address Register

ITDEDADR_PE1 register is used to hold the address when a DED error is detected. Error address capture trigger is same as corresponding DEDF set in ITDERSTR_PE1. Entry address is stored in DEADR[7:0] and way group number in DEADR[8] in this register. ITDEDADR_PE1 is cleared by setting DSTCLR in ITDERSTCLR_PE1 register to 1.

Access: ITDEDADR_PE1 can be read in 32-bit units.

Address: FFC6 14E0_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | DEADR[8:0] | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.75 ITDEDADR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 9 | Reserved | When read, the value after reset is read. |
| 8 to 0 | DEADR[8:0] | 2-bit error detection address |

31.2.6.4 Test Function

A cache instruction is used to write the desired values to RAM data and the ECC bits, and read data in the RAM and the ECC bits directly.

Since instructions as described above go through the same encoding or decoding path as a normal cache fill or instruction fetch, only such instructions can be used in inserting and confirming errors.

For details, see *RH850G3M User's Manual: Software*.

31.2.7 DTS RAM ECC

See **Section 7, DMA**.

31.2.8 ECC for Peripheral RAM (32 Bits)

31.2.8.1 Overview

This is an ECC module for the RAM of the following peripheral modules.

FlexRay, CSIHn (n = 0 to 3), CAN

Error Detection and Correction

Seven-bit ECC data is appended to the 32-bit RAM data.

This ECC circuit provides ECC 2-bit error detection and ECC 1-bit error detection and correction.

Enabling or Disabling ECC Error Detection and Correction

- ECC error detection can be either enabled or disabled.
- 1-bit ECC error correction can be either enabled or disabled.
- If all the bits of RAM output data are stuck to 0 or 1, it is detected as an ECC 2-bit error.

ECM error notifications

- An error notification is issued upon detection of an ECC 2-bit error (notification can be either enabled or disabled).
- An error notification is issued upon detection of an ECC 1-bit error (notification can be either enabled or disabled).
- An error notification is issued upon detection of an address buffer overflow error for ECC error.

Once an error notification is issued, another error notification is not issued until the corresponding error status is cleared even if another ECC error is detected.

Error Status

- Detection of ECC 2- and 1-bit errors can be monitored.
- Special registers are provided to clear error status.

Address Capture

- Only one address at which an ECC error has occurred can be captured.
- A signal is generated upon detection of ECC 2-bit or 1-bit error, and the signal is used as a trigger to capture the error-causing address (when the first (1-bit or 2-bit) error is detected after the flag is cleared).

Testing Function (Error Injection)

- By setting the test mode, register values can be used as the data to be output to the RAM. When a peripheral module writes to the RAM, the ECEDB[31:0] register value can be written to the RAM data section, and the ERDB[6:0] register value can be written to the ECC bit section.
- By setting the test mode, the ECC bit section can be latched when RAM data is read, and the value can be confirmed.
- By setting the test mode, the ECC bit (encoding circuit) and syndrome code (decoding circuit), which are generated from the input data, can be confirmed.

31.2.8.2 List of Registers

(1) List of ECC Modules

ECC modules are provided for RAMs of multiple peripheral functions. The following table shows the peripheral functions provided with the ECC modules, the corresponding ECC module names, and base addresses of the ECC modules.

Table 31.76 ECC Module List

| Peripheral Functions | | Symbol | ECC Module Names and Register Base Addresses | |
|----------------------|-----------------------------------|-----------|--|--------------------------|
| | | | Module Name | Base Address <base_addr> |
| FlexRay | Message RAM (MRAM) | ECCFLX0 | ECCFLX0 | FFC7 2000 _H |
| | Temporary buffer (TBF A) | ECCFLX0T1 | ECCFLX0T1 | FFC7 2040 _H |
| | Temporary buffer (TBF B) | ECCFLX0T0 | ECCFLX0T0 | FFC7 2080 _H |
| CSIH | CSIH0 | ECCCSIH0 | ECCCSIH0 | FFC7 0000 _H |
| | CSIH1 | ECCCSIH1 | ECCCSIH1 | FFC7 0040 _H |
| | CSIH2 | ECCCSIH2 | ECCCSIH2 | FFC7 0080 _H |
| | CSIH3 | ECCCSIH3 | ECCCSIH3 | FFC7 00C0 _H |
| RS-CANFD | RS-CANFD (Buffer RAM) | ECCCAN0 | ECCCAN0 | FFC7 1100 _H |
| | RS-CANFD (Receive rule table RAM) | ECCCAN1 | ECCCAN1 | FFC7 1200 _H |

(2) List of Registers

Each ECC module has the registers shown in the following table.

Table 31.77 List of Registers

| Register Name | Additional Abbreviation*2 | R/W | Value after reset | Address | Access Size |
|--|---------------------------|-----|------------------------|-------------------------------|-------------|
| ECC control register*1 | CTL | R/W | 0000 001X _H | <base_addr> + 00 _H | 32 |
| ECC test mode control register | TMC | R/W | 0000 0000 _H | <base_addr> + 04 _H | 32 |
| ECC bit data control test register | TRC | R/W | 0000 0000 _H | <base_addr> + 08 _H | 32 |
| ECC encoder and decoder data test register | TED | R/W | 0000 0000 _H | <base_addr> + 0C _H | 32 |
| ECC error address register | EAD0 | R | 0000 0000 _H | <base_addr> + 10 _H | 32 |

Note 1. The reset value of the LSB in the ECC control register is undefined.

Note 2. "Additional Abbreviation" is added to the symbol in the list of ECC modules that correspond to peripheral functions. For example, ECCCSIH2TMC represents the ECC test mode control register of CSIH2.

(3) Register Map**Table 31.78 Register Map**

| Abbreviation | 31 | 24 | 23 | 16 | 15 | 8 | 7 | 0 | Address |
|--------------|----------------------|--------------|----------------------|--------------|-----------|-------------|----------|------------|-------------------|
| CTL | — (00 _H) | | CTL[17:16] | | CTL[15:8] | | CTL[7:0] | | nn00 _H |
| TMC | — (00 _H) | | — (00 _H) | | TMC[15:8] | | TMC[7:0] | | nn04 _H |
| TRC | | SYND[7:0] | | HORD[7:0] | | ECRD[7:0] | | ERDB[7:0] | nn08 _H |
| TED | | ECEDB[31:24] | | ECEDB[23:16] | | ECEDB[15:8] | | ECEDB[7:0] | nn0C _H |
| EAD0 | | ECEAD[31:24] | | ECEAD[23:16] | | ECEAD[15:8] | | ECEAD[7:0] | nn10 _H |

31.2.8.3 Details of Registers

(1) CTL — ECC Control Register

The CTL register controls the mode of the ECC for target peripheral modules.

Bits 7, 5, 4 and 3 should be set (written) while the target peripheral module operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: CTL can be read/written in 32-bit units.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: Undefined

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|----|----|--------|--------|--------|----|-------|----|--------|---------|---------|--------|---------|-----------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ECDEDF0 | ECSEDF0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EMCA1 | EMCA0 | — | — | ECOVFF | ECER2C | ECER1C | — | ECTHM | — | EC1ECP | EC2EDIC | EC1EDIC | ECER2F | ECER1F | ECEMF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Undefined |
| R/W | R/W*1 | R/W*1 | R | R | R | R/W*1 | R/W*1 | R | R/W | R | R/W | R/W | R/W | R | R | R |

Note 1. This bit is always read as 0.

Table 31.79 CTL Register Contents (1/3)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|----------|--|---------|---------|-----------------------|---|---|---|---|---|---|---|---|---|---|---|-------------|
| 31 to 18 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |
| 17 | ECDEDF0 | | | | | | | | | | | | | | | | |
| 16 | ECSEDF0 | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>ECDEDF0</th><th>ECSEDF0</th><th>Operation description</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>There is no error address in EAD0 after reset or clearing of ECER2F and ECER1F bits These bits are cleared by: (1) reset (2) writing ECER2C = 1 or ECER1C = 1 (3) enabling through mode (ECTHM = 1)</td></tr> <tr> <td>1</td><td>0</td><td>Address captured in EAD0 shows that ECC 2-bit error has occurred and the related address is captured.</td></tr> <tr> <td>0</td><td>1</td><td>Address captured in EAD0 shows that ECC 1-bit error has occurred and the related address is captured.</td></tr> <tr> <td>1</td><td>1</td><td>Not defined</td></tr> </tbody> </table> | ECDEDF0 | ECSEDF0 | Operation description | 0 | 0 | There is no error address in EAD0 after reset or clearing of ECER2F and ECER1F bits These bits are cleared by: (1) reset (2) writing ECER2C = 1 or ECER1C = 1 (3) enabling through mode (ECTHM = 1) | 1 | 0 | Address captured in EAD0 shows that ECC 2-bit error has occurred and the related address is captured. | 0 | 1 | Address captured in EAD0 shows that ECC 1-bit error has occurred and the related address is captured. | 1 | 1 | Not defined |
| ECDEDF0 | ECSEDF0 | Operation description | | | | | | | | | | | | | | | |
| 0 | 0 | There is no error address in EAD0 after reset or clearing of ECER2F and ECER1F bits These bits are cleared by: (1) reset (2) writing ECER2C = 1 or ECER1C = 1 (3) enabling through mode (ECTHM = 1) | | | | | | | | | | | | | | | |
| 1 | 0 | Address captured in EAD0 shows that ECC 2-bit error has occurred and the related address is captured. | | | | | | | | | | | | | | | |
| 0 | 1 | Address captured in EAD0 shows that ECC 1-bit error has occurred and the related address is captured. | | | | | | | | | | | | | | | |
| 1 | 1 | Not defined | | | | | | | | | | | | | | | |
| 15 | EMCA1 | Access control bits 1 and 0 to ECC mode selection bit | | | | | | | | | | | | | | | |
| 14 | EMCA0 | These bits specify whether updating the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01 _B , writing to bit 7 is enabled. | | | | | | | | | | | | | | | |
| 13, 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | | | | | | | | | | |

Table 31.79 CTL Register Contents (2/3)

| Bit Position | Bit Name | Function | | | | | | |
|--------------|--|--|--------|-----------------------|---|--|---|------------------------------------|
| 11 | ECOVFF | When an error is detected while the error status is set and the address of the new error is different from the address that is already latched (not cleared or reset is not issued), this bit is set and error notification is generated. <table border="1"> <thead> <tr> <th>ECOVFF</th> <th>Operation description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Overflow has not occurred after reset or clearing of ECER2F and ECER1F. These bits are cleared by: (1) reset (2) writing ECER2C = 1 or ECRE1C = 1. (3) enabling through mode (ECTHM = 1)</td> </tr> <tr> <td>1</td> <td>Error address register overflowed.</td> </tr> </tbody> </table> | ECOVFF | Operation description | 0 | Overflow has not occurred after reset or clearing of ECER2F and ECER1F. These bits are cleared by: (1) reset (2) writing ECER2C = 1 or ECRE1C = 1. (3) enabling through mode (ECTHM = 1) | 1 | Error address register overflowed. |
| ECOVFF | Operation description | | | | | | | |
| 0 | Overflow has not occurred after reset or clearing of ECER2F and ECER1F. These bits are cleared by: (1) reset (2) writing ECER2C = 1 or ECRE1C = 1. (3) enabling through mode (ECTHM = 1) | | | | | | | |
| 1 | Error address register overflowed. | | | | | | | |
| 10 | ECER2C | ECC 2-bit error detection flag clear bit This bit is used to clear bit 2, the status flag (ECER2F). This bit is always read as 0. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between writing 0 to this bit and setting the ECER2F bit occurs, writing to this bit has priority. | | | | | | |
| 9 | ECER1C | ECC 1-bit error detection correction accumulation flag clear bit This bit is used to clear bit 1, the status flag (ECER1F). This bit is always read as 0. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between writing 0 to this bit and setting the ECER1F bit occurs, writing to this bit has priority. | | | | | | |
| 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | |
| 7 | ECTHM | ECC Function Through Mode Selection bit Set this bit to select whether to pass through the function of the ECC decoder. When writing to this bit, (0, 1) should be written to (EMCA1, EMCA0) at the same time. Set this bit to 1 to disable the ECC function. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled. The encoder is not affected. The decoder does not conduct error detection and bit correction. | | | | | | |
| 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. | | | | | | |
| 5 | EC1ECP | ECC 1-bit error correction enable bit This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected. | | | | | | |
| 4 | EC2EDIC | ECC 2-bit error detection notification enable bit This bit controls whether to notify a 2-bit error to ECM when it is detected. 0: When a 2-bit error is detected, it is not notified to ECM. 1: When a 2-bit error is detected, it is notified to ECM. | | | | | | |
| 3 | EC1EDIC | ECC 1-bit error detection notification enable bit This bit controls whether to notify a 1-bit error to ECM when it is detected. 0: When a 1-bit error is detected, it is not notified to ECM. 1: When a 1-bit error is detected, it is notified to ECM. | | | | | | |
| 2 | ECER2F | ECC 2-bit error detection flag bit This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled. When 2-bit error notification is enabled and this flag is set, a 2-bit error notification signal is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. If 2-bit error is detected again while this bit is set, an error notification request will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred. | | | | | | |

Table 31.79 CTL Register Contents (3/3)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 1 | ECER1F | <p>1-bit error detection/correction flag bit</p> <p>This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled. Write 1 to the ECER1C bit (bit 9) to clear the flag.</p> <p>0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred.</p> |
| 0 | ECEMF | <p>ECC error message flag</p> <p>This flag indicates whether an error exists in the current read data bus. This bit is updated whenever the RAM outputs data. Because the value after reset of the RAM data is undefined, if the RAM is read before initialization, this bit may be set.</p> <p>0: The current RAM output data does not have bit errors. 1: The current RAM output data has bit errors.</p> |

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.
We recommend initializing the RAM before clearing bits 2 and 1.

(2) TMC — ECC Test Mode Control Register

The TMC register is used to control the test mode.

When writing to bit 7, ETMA1 and ETMA0 need to be 10_B.

Access: TMC can be read/written in 32-bit units.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|--------|----|----|--------|--------|-------|-------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ETMA1 | ETMA0 | — | — | — | — | — | — | ECTMCE | — | — | ECTRRS | ECREOS | ECENS | ECDCS | ECREIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W*1 | R/W*1 | R | R | R | R | R | R | R/W | R | R | R/W | R/W | R/W | R/W | R/W |

Note 1. This bit is always read as 0.

Table 31.80 TMC Register Contents (1/2)

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 | ETMA1 | Access control bit 1 and 0 to ECC test mode bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10 _B , writing to bit 7 is enabled. |
| 14 | ETMA0 | |
| 13 to 8 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 7 | ECTMCE | ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, (1, 0) should be written to (ETMA1, ETMA0) at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. |
| 6, 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 | ECTRRS | ECC RAM read test mode selection bit This bit is used to generate a RAM read cycle. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: A RAM read cycle is not generated even when TED is read. 1: A RAM read cycle is generated when TED is read. In addition, the TED read value depends on the setting of ECDCS bit (bit 1). The ERDB read value depends on the setting of ECREIS bit (bit 0). |
| 3 | ECREOS | ECC bit output data selection bit This bit specifies which is output as the ECC bit output, the ECC encoder output data or the value of the ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Encoding result is output as the ECC bit output. 1: TRC.ERDB[6:0] is output as the ECC bit output. |

Table 31.80 TMC Register Contents (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 2 | ECENS | <p>ECC encoder input selection bit</p> <p>This bit specifies which is input to the encoder, the data from the peripheral macro or the value of the TED.ECEDB[31:0]. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: Write data from the peripheral macro to the RAM is input to the ECC encoder as input data.</p> <p>1: TED.ECEDB[31:0] is input to the ECC encoder as input data.</p> |
| 1 | ECDCS | <p>ECC decoder input selection bit</p> <p>This bit specifies which is input to the decoder as the lower 32-bit data of the decoder input, the lower 32-bit data from RAM or the value of TED.ECEDB[31:0]. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: The lower 32-bit data from the RAM output data is input to the data area (lower 32-bit data) to the decoder circuit.</p> <p>1: TED.ECEDB[31:0] is input to the data area to the decoder circuit.</p> |
| 0 | ECREIS | <p>ECC bit input data selection bit</p> <p>This bit specifies which is input as the upper 7-bit data of the decoder input, the upper 7-bit data (redundant bit area) from the RAM or the value of the TRC.ERDB[6:0]. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: Upper 7 bits of RAM output data are input to the ECC redundant bit area to the decoder circuit.</p> <p>1: TRC.ERDB[6:0] is input to the ECC redundant bit area to the decoder circuit.</p> |

(3) TED — ECC Encode and Decode Data Test Register

TED is a test register for 32-bit data used for ECC encoding/decoding.

In test mode, the value of this register is used as input data for the encoder or decoder circuit.

Access: When TMC.ECTMCE = 1, TED can be read/written in 32-bit units.
When TMC.ECTMCE = 0, the value of TED is always 0000_H.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | ECEDB[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECEDB[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.81 TED Register Contents

| Bit position | Bit Name | Function |
|--------------|-------------|--|
| 31 to 0 | ECEDB[31:0] | When TMC.ECENS = 1, the value of this register is used as the input data to the encoder circuit and sent to the RAM. When TMC.ECDCS = 1, the value of this register is used as the bit 31-0 of the input data to the decoder circuit. In addition, when TMC.ECTRRS = 1, the read value from this register switches from the value of this register to the RAM output data. |

(4) TRC — ECC Redundant Bit Data Control Test Register

This register is a 32-bit test register for the ECC redundant bit area and consists of four 8-bit registers, SYND, HORD, ECRD, and ERDB.

Access: When TMC.ECTMCE = 1, TRC can be read/written in 32-bit units.
When TMC.ECTMCE = 0, the value of TRC is always 0000_H.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----------------|----|----|----|----|----|----|----|----------------|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | SYND (See (5)) | | | | | | | | HORD (See (6)) | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECRD (See (7)) | | | | | | | | ERDB (See (8)) | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

(5) SYND — ECC Decoder Syndrome Data Register

SYND is a read-only register to confirm the syndrome code generated by the decoding circuit in test mode (ECTMCE = 1).

Write access to SYND is ignored.

Access: When TMC.ECTMCE = 1, SYND can be read only in 8-bit units.
When TMC.ECTMCE = 0, the value of SYND is always 00_H.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | SYND6 | SYND5 | SYND4 | SYND3 | SYND2 | SYND1 | SYND0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 31.82 SYND Register Contents

| Bit position | Bit Name | Function |
|--------------|-----------|---|
| 7 | Reserved | When read, write the value after reset is read. |
| 6 to 0 | SYND[6:0] | When reading these bits, the syndrome code (SYND[6:0]) generated in the decoder circuit based on the input data can be read. The value of these bits changes as the input data changes. Note that this register is enabled only when ECTMCE = 1, and the value is always 00 _H when ECTMCE = 0. |

(6) HORD — ECC 7-Bit Redundant Bit Data Holding Test Register

HORD holds the 7-bit ECC redundant area (upper 7-bit of RAM data), which cannot be confirmed by the peripheral module when it reads the RAM in test mode (ECTMCE = 1).

Access: When TMC.ECTMCE = 1, HORD can be read only in 8-bit units.
When TMC.ECTMCE = 0, the value of HORD is always 00_H.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|-------|-------|-------|-------|-------|-------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | HORD6 | HORD5 | HORD4 | HORD3 | HORD2 | HORD1 | HORD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 31.83 HORD Register Contents

| Bit position | Bit Name | Function |
|--------------|-----------|--|
| 7 | Reserved | When read, write the value after reset is read. |
| 6 to 0 | HORD[6:0] | When the peripheral module reads the RAM in test mode (ECTMCE = 1), this register fetches the upper 7 bits of the RAM output data at the rising edge of the operating clock. In addition, if TMC.ECTRRS = 1, the value of the RAM output data is fetched to this register when the ECEDB[15:0] bits are read. Note that this register is enabled only when ECTMCE = 1. |

(7) ECRD — ECC Encoder Test Register

ECRD is a read-only register to read the 7-bit redundant section generated by the encoding circuit in test mode (ECTMCE = 1).

Access: When TMC.ECTMCE = 1, ECRD can be read only in 8-bit units.
When TMC.ECTMCE = 0, the value of ECRD is always 00_H.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|-------|-------|-------|-------|-------|-------|-------|
| | — | ECRD6 | ECRD5 | ECRD4 | ECRD3 | ECRD2 | ECRD1 | ECRD0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 31.84 ECRD Register Contents

| Bit position | Bit Name | Function |
|--------------|-----------|--|
| 7 | Reserved | When read, write the value after reset is read. |
| 6 to 0 | ECRD[6:0] | ECRD is used to hold the ECC bits generated from the input data of the peripheral module. These bits can be checked to confirm correct encoding. The read value is the encoding result (ECC[6:0]), and not the ECC redundant bit output value. Note that this register is enabled only when ECTMCE = 1. |

(8) ERDB — ECC Bit Input and Output Substitution Buffer Register

ERDB is a buffer register for the data that substitutes for the input and output data for the 7-bit ECC redundant data area in test mode (ECTMCE = 1).

ERDB can be read and written to in ECC test mode (ECTMCE = 1).

Access: When TMC.ECTMCE = 1, ERDB can be read/written in 8-bit units.
When TMC.ECTMCE = 0, the value of ERDB is always 00_H.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|-------|-------|-------|-------|-------|-------|-------|
| | — | ERDB6 | ERDB5 | ERDB4 | ERDB3 | ERDB2 | ERDB1 | ERDB0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.85 ERDB Register Contents

| Bit position | Bit Name | Function |
|--------------|-----------|---|
| 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 to 0 | ERDB[6:0] | When ECREOS = 1, the value of this register is output to the ECC redundant bit output pins and sent to the RAM instead of the 7-bit long redundant bits generated by the encoder circuit. When ECREIS = 1, the value of this register is used by the decoder circuit instead of the upper 7 bits of the data input to the decoder circuit. In addition, when ECTRRS = 1, the read value from this register switches from the written value to this register to the RAM output data. |

(9) EAD0 — ECC Error Address Register

EAD0 is a read-only register to hold the address at which an ECC error has occurred.

Access: EAD0 can be read in 32-bit units.

Address: See Table 31.76, ECC Module List and Table 31.78, Register Map.

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | ECEAD[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECEAD[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.86 EAD0 Register Contents

| Bit position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 0 | ECEAD[31:0] | <p>EAD0 is a read-only register to hold the address at which an ECC error has occurred.</p> <p>If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in EAD0 as the address at which the ECC error has occurred.</p> <p>The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored.</p> <p>Only one address can be held in EAD0.</p> |

31.2.8.4 Test Function

(1) Writing RAM data

Write data to the peripheral RAM. Note that ECC corresponding to the write data is simultaneously written to the ECC bits. To write an arbitrary value to the ECC bits, use ECC test mode shown in (3).

(2) Reading RAM data

- (a) Set the ECTHM bit in the ECC control register to 1 to disable ECC error detection/correction.
- (b) Read the peripheral RAM data. The RAM data is directly read because error detection or correction is not performed during reading.

Exiting this test mode

- (a) Set the ECTHM bit in the ECC control register to 0 to enable ECC error detection/correction.

(3) Writing to ECC bits

- (a) Set the ECTMCE bit in the ECC test mode control register to 1 to specify ECC test mode.
- (b) Write a value to be written to the ECC bits to TRC.ERDB[6:0].
- (c) Set the ECREOS bit in the ECC test mode control register to 1 to select TRC.ERDB[6:0] to be written to the ECC bits.
- (d) When data is written to the peripheral RAM, the TRC.ERDB[6:0] value is written to the ECC bits.

Exiting this test mode

- (a) Set the ECTMCE bit in the ECC test mode control register to 0 to specify normal mode.

(4) Reading ECC bits

- (a) Set the ECTMCE bit in the ECC test mode control register to 1 to specify ECC test mode.
- (b) When the peripheral RAM data is read, the value of the ECC bits is stored in TRC.HORD[6:0].

Exiting this test mode

- (a) Set the ECTMCE bit in the ECC test mode control register to 0 to specify normal mode.

31.2.9 ECC on Data Transfer Path

31.2.9.1 ECC protection on Bus

The target paths of ECC protection on bus are shown in the following table. It is possible to detect address and data line errors from an access master to an access slave. If ECC error is detected, an error signal is sent to ECM.

Table 31.87 ECC protection on Bus

| Access Source (Master) | Access Destination (Slave) |
|------------------------|---|
| CPU1, DMAC, DTS | RAM of CPU1, Global RAM, all P-bus groups |

Data protection targets are provided with ECC decoder controllers which can be used to configure settings such as detection and signaling. Error information is stored in status registers when error occurs.

In the case of an address error detection, even if detected error is 1 bit error, MCU indicates that a serious unrecoverable problem has occurred.

31.2.9.2 Failure Detection Function for GRAM Access

The GRAM IF (also known as GRAM Controller (GRAMC)) is comprised of the Write-Through-Buffer (WT-Buffer) for performance enhancements, the Read-Modify-Write (RmW) to allow various access bit-width (64/32/16/8 bit) and the arbiter for arbitration of several masters. Refer to the description of GRAM access in **Section 3, CPU System** for details on user function.

Here, the safety mechanisms of the GRAMC are described:

1. WT-Buffer: The data in WT-Buffer is protected by data ECC, and other logics such as Control, Address and Vld are made redundant.
2. Address path: The address path of the data is protected by address ECC. The address ECC is generated inside the PE and evaluated at output of GRAMC.
3. RmW: For 8-bit and 16-bit write accesses, the data ECC has to be recomputed, hence the Read-Modify-Write circuits are redundantly implemented. The RmW circuit before the arbiter does not include a state machine, therefore the RmW state checker is implemented for the RmW circuit after the arbiter.
4. Request buffers: The request buffers are protected by address ECC and data ECC.
5. Arbiter: The arbiter is redundantly implemented.
6. Response buffers: The response buffers are protected by data ECC and the packet ID. If the ID of request packet doesn't match with the ID of response packet, GRAM master cannot proceed to the next process. As a result, the failure is detected by watch dog timer timeout or E2E communication.

31.2.9.3 Fault Detection Function of Arbitration

Bus arbiters in PFSS have arbiter check function which detects unintended arbiter status. Detected arbitration error is notified to ECM. Target arbiters are as follow:

Table 31.88 Arbiter Check

| Target arbiter | Function |
|---------------------|----------------------------------|
| GRAMC | Global RAM arbiter is redundant. |
| FABTSS | Code Flash arbiter is redundant. |
| System interconnect | Arbitration check function |

31.2.9.4 List of Registers

Table 31.89 List of Registers (1/4)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|--------------------|---|-----|------------------------|-------------|
| FFC62C00 _H | CFECCCTL_VCI2CFBB | Code Flash Address ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC62C04 _H | CFERRINT_VCI2CFBB | Code Flash Address Error Information Control Register | R/W | 0000 0030 _H | 32/16/8 |
| FFC62C08 _H | CFERSTCLR_VCI2CFBB | Code Flash Address ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC62C0C _H | CFOVFSTR_VCI2CFBB | Code Flash Address Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC62C10 _H | CFERSTR_VCI2CFBB | Code Flash Address ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC62C50 _H | CFEADR0_VCI2CFBB | Code Flash Address ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC63400 _H | IFECCCTL_PE1 | IFU Data ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC63404 _H | IFERRINT_PE1 | IFU Data Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC63408 _H | IFERSTCLR_PE1 | IFU Data ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6340C _H | IFOVFSTR_PE1 | IFU Data Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC63410 _H | IFERSTR_PE1 | IFU Data ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC63450 _H | IFERADR_PE1 | IFU Data ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC66000 _H | LSSECCCTL_PE1 | LSU Slave Data and Address ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC66004 _H | LSSERRINT_PE1 | LSU Slave Data and Address Error Information Control Register | R/W | 0000 0033 _H | 32/16/8 |
| FFC66008 _H | LSSERSTCLR_PE1 | LSU Slave Data and Address ECC SED/DED Status Clear Register | R/W | 0000 0000 _H | 32/16/8 |
| FFC6600C _H | LSSOVFSTR_PE1 | LSU Slave Data and Address Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |

Table 31.89 List of Registers (2/4)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|----------------|--|-----|------------------------|-------------|
| FFC66010 _H | LSSERSTR_PE1 | LSU Slave Data and Address ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC66050 _H | LSSEADR_PE1 | LSU Slave Data and Address ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC66400 _H | LSMECCCTL_PE1 | LSU Master Data ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC66404 _H | LSMERRINT_PE1 | LSU Master Data Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC66408 _H | LSMERSTCLR_PE1 | LSU Master Data ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6640C _H | LSMOVFSTR_PE1 | LSU Master Data Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC66410 _H | LSMERSTR_PE1 | LSU Master Data ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC66450 _H | LSMEADR_PE1 | LSU Master Data ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC67004 _H | VPERRINT_SG0 | System interconnect (PBus I/F) Data Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC67008 _H | VPERSTCLR_SG0 | System interconnect (PBus I/F) Data ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6700C _H | VPOVFSTR_SG0 | System interconnect (PBus I/F) Data Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC67010 _H | VPERSTR_SG0 | System interconnect (PBus I/F) Data ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC67050 _H | VPEADR_SG0 | System interconnect (PBus I/F) Data ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC67404 _H | VPERRINT_SG1 | System interconnect (PBus I/F) Data Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC67408 _H | VPERSTCLR_SG1 | System interconnect (PBus I/F) Data ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6740C _H | VPOVFSTR_SG1 | System interconnect (PBus I/F) Data Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC67410 _H | VPERSTR_SG1 | System interconnect (PBus I/F) Data ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC67450 _H | VPEADR_SG1 | System interconnect (PBus I/F) Data ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC67804 _H | VPERRINT_SG2 | System interconnect (PBus I/F) Data Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC67808 _H | VPERSTCLR_SG2 | System interconnect (PBus I/F) Data ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |

Table 31.89 List of Registers (3/4)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|------------------------|----------------|---|-----|------------------------|-------------|
| FFC6780C _H | VPOVFSTR_SG2 | System interconnect (PBus I/ F) Data Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC67810 _H | VPERSTR_SG2 | System interconnect (PBus I/ F) Data ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC67850 _H | VPEADR_SG2 | System interconnect (PBus I/ F) Data ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC68404 _H | VPERRINT_SG5 | System interconnect (PBus I/ F) Data Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC68408 _H | VPERSTCLR_SG5 | System interconnect (PBus I/ F) Data ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6840C _H | VPOVFSTR_SG5 | System interconnect (PBus I/ F) Data Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC68410 _H | VPERSTR_SG5 | System interconnect (PBus I/ F) Data ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC68450 _H | VPEADR_SG5 | System interconnect (PBus I/ F) Data ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC6A000 _H | VCECCCTL_PDMA | System Interconnect Data ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC6A004 _H | VCERRINT_PDMA | System Interconnect Error Information Control Register | R/W | 0000 0003 _H | 32/16/8 |
| FFC6A008 _H | VCERSTCLR_PDMA | System Interconnect Data ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6A00C _H | VCOVFSTR_PDMA | System Interconnect Data Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC6A010 _H | VCERSTR_PDMA | System Interconnect Data ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC6A050 _H | VCEADR_PDMA | System Interconnect Data ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC6C000 _H | APECCCTL_PFSS | P-bus Data and Address ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC6C004 _H | APERRINT_PFSS | P-bus Error Information Control Register | R/W | 0000 0073 _H | 32/16/8 |
| FFC6C008 _H | APERSTCLR_PFSS | P-bus Data and Address ECC SED/DED Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC6C00C _H | APOVFSTR_PFSS | P-bus Data and Address Error Count Overflow Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC6C010 _H | APERSTR_PFSS | P-bus Data and Address ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC6C050 _H | APEADR_PFSS | P-bus Data and Address ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFDDDD800 _H | APEC0ECCCTL | ECC Control Register | R/W | 0000 0000 _H | 32/16 |

Table 31.89 List of Registers (4/4)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|-----------------------|---------------|---|-----|------------------------|-------------|
| FFDDD804 _H | APEC0ERRINT | ECC Error Information Register | R/W | 0000 0073 _H | 32/16/8 |
| FFDDD808 _H | APEC0STCLR | Error Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFDDD80C _H | APEC0OVFSTR | Error Overflow Register | R | 0000 0000 _H | 32/16/8 |
| FFDDD810 _H | APEC01STERSTR | P-bus Data and Address ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFDDD850 _H | APEC01STEADR0 | P-bus Data and Address ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFF94800 _H | APEC1ECCCTL | ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFF94804 _H | APEC1ERRINT | ECC Error Information Register | R/W | 0000 0073 _H | 32/16/8 |
| FFF94808 _H | APEC1STCLR | Error Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFF9480C _H | APEC1OVFSTR | Error Overflow Register | R | 0000 0000 _H | 32/16/8 |
| FFF94810 _H | APEC11STERSTR | P-bus Data and Address ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFF94850 _H | APEC11STEADR0 | P-bus Data and Address ECC SED/DED Address Register | R | 0000 0000 _H | 32 |
| FFC40800 _H | APEC2ECCCTL | ECC Control Register | R/W | 0000 0000 _H | 32/16 |
| FFC40804 _H | APEC2ERRINT | ECC Error Information Register | R/W | 0000 0073 _H | 32/16/8 |
| FFC40808 _H | APEC2STCLR | Error Status Clear Register | W | 0000 0000 _H | 32/16/8 |
| FFC4080C _H | APEC2OVFSTR | Error Overflow Register | R | 0000 0000 _H | 32/16/8 |
| FFC40810 _H | APEC21STERSTR | P-bus Data and Address ECC SED/DED Status Register | R | 0000 0000 _H | 32/16/8 |
| FFC40850 _H | APEC21STEADR0 | P-bus Data and Address ECC SED/DED Address Register | R | 0000 0000 _H | 32 |

31.2.9.5 Details of Registers

(1) CFECCTL_VCI2CFBB — Code Flash Address ECC Control Register

CFECCTL_VCI2CFBB register controls the address ECC error detection/correction and 1-bit error correction on bus width conversion between code flash interface and system interconnection. Writing ECC control registers must be executed with $PROT[1:0] = 01_B$.

Access: CFECCTL_VCI2CFBB can be read/written in 32/16-bit units.

Address: FFC6 2C00_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | ASECDIS | AECCLDIS | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R |

Table 31.90 CFECCTL_VCI2CFBB Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | ASECDIS | Address ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Address ECC error detection/correction (AECCLDIS = 0). 0: Correction is carried out when 1-bit is detected 1: Correction is not carried out when 1-bit is detected |
| 2 | AECCLDIS | Address ECC disable bit Enables or disables Address ECC error detection/correction. 0: Address ECC error detection/correction is enabled 1: Address ECC error detection/correction is disabled |
| 1, 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(2) CFERRINT_VCI2CFBB — Code Flash Address Error Information Control Register

CFERRINT_VCI2CFBB register controls whether error information is reported to ECM, when address ECC 2-bit error and address ECC 1-bit error are detected.

Access: CFERRINT_VCI2CFBB can be read/written in 32/16/8-bit units.

Address: FFC6 2C04_H

Value after reset: 0000 0030_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|------------|--------|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | ADEDI E | ASEDIE | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R | R | R |

Table 31.91 CFERRINT_VCI2CFBB Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | ADEDIE | Address ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled |
| 4 | ASEDIE | Address ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled |
| 3 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(3) CFERSTCLR_VCI2CFBB — Code Flash Address ECC SED/DED Status Clear Register

CFERSTCLR_VCI2CFBB register is used to clear error flag in CFERSTR_VCI2CFBB, error overflow flag in CFOVFSTR_VCI2CFBB, and error address in CFEADR0_VCI2CFBB. This is write only register and read value is always “0”.

Access: CFERSTCLR_VCI2CFBB can be written only in 32/16/8-bit units.

Address: FFC6 2C08_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.92 CFERSTCLR_VCI2CFBB Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | Error flag clear Writing “1” to this bit clears ADED/ASEDF in CFERSTR_VCI2CFBB and ERROVF in CFOVFSTR_VCI2CFBB and CFEADR0_VCI2CFBB. |

(4) CFOVFSTR_VCI2CFBB — Code Flash Address Error Count Overflow Status Register

CFOVFSTR_VCI2CFBB register monitors occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting SSTCLR in CFERSTCLR_VCI2CFBB register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: CFOVFSTR_VCI2CFBB can be read only in 32/16/8-bit units.

Address: FFC6 2C0C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.93 CFOVFSTR_VCI2CFBB Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error overflow flag If an error flag (ADED/ASEDF in CFERSTR_VCI2CFBB) is set and a different error is detected, this bit is set. |

(5) CFERSTR_VCI2CFBB — Code Flash Address ECC SED/DED Status Register

CFERSTR_VCI2CFBB is the address error monitor register. When all error flags are “0” for each bank and a new error occurs, the error status flag is set. If address ECC 1-bit error monitor flag is set and the new error is address ECC 2 bit error, the new error is set (does not clear the previous error flag). If multiple error causes are simultaneously detected, detected errors are all set (e.g. if ADED is detected at the same access, ADEDF is set.). CFERSTR_VCI2CFBB register is cleared by setting SSTCLR in CFERSTCLR_VCI2CFBB register to 1.

Access: CFERSTR_VCI2CFBB can be read only in 32/16/8-bit units.

Address: FFC6 2C10_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-------|-------|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | ADEDF | ASEDF | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.94 CFERSTR_VCI2CFBB Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 8 | Reserved | When read, the value after reset is read. |
| 7 | ADEDF | Address ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in CFERSTCLR_VCI2CFBB Condition for “1”: ADEDF is “0” and ECC 2-bit error is detected. |
| 6 | ASEDF | Address ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in CFERSTCLR_VCI2CFBB Condition for “1”: ASEDF is “0” and ECC 1-bit error is detected. |
| 5 to 0 | Reserved | When read, the value after reset is read. |

(6) CFEADR0_VCI2CFBB — Code Flash Address ECC SED/DED Address Register

CFEADR0_VCI2CFBB register is used to hold the address when all error flags are not set and an error is detected, or, if ASEDF in CFERSTR_VCI2CFBB is set and address ECC 2-bit error is detected. This register stores a corrected address in case of 1-bit error when address correction is enabled. But when address correction is disabled or address has a bit error which exceeds 2 bits, the address with error is stored.

CFEADR0_VCI2CFBB is cleared by setting SSTCLR in CFERSTCLR_VCI2CFBB register to 1.

Access: CFEADR0_VCI2CFBB can be read only in 32-bit units.

Address: FFC6 2C50_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| EADR[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EADR[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.95 CFEADR0_VCI2CFBB Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

(7) IFECCTL_PE1 — IFU Data ECC Control Register

IFECCTL_PE1 register controls the ECC error detection/correction and 1-bit error correction on master port in instruction fetch unit that is used to fetch an instruction to global or local RAM. Writing ECC control registers must be executed with PROT[1:0] = 01.

Access: IFECCTL_PE1 can be read/written in 32/16-bit units.

Address: FFC6 3400_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.96 IFECCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15 | PROT1 | Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to ITECCCTL_PE1. |
| 14 | PROT0 | |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | 1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Set (PROT1, PROT0) = (0, 1) simultaneously when writing to this bit. 0: Enables correction of the 1-bit error detected. 1: Disables correction of the 1-bit error detected. |
| 0 | ECCDIS | ECC Disable Enables or disables ECC error detection and correction. Set (PROT1, PROT0) = (0, 1) simultaneously when writing to this bit. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction. |

(8) IFERRINT_PE1 — IFU Data Error Information Control Register

IFERRINT_PE1 register controls whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected.

Access: IFERRINT_PE1 can be read/written in 32/16/8-bit units.

Address: FFC6 3404_H

Value after reset: 0000 0003_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.97 IFERRINT_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | ECC 2-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error. |
| 0 | SEDIE | ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error. |

(9) IFERSTCLR_PE1 — IFU Data ECC SED/DED Status Clear Register

IFERSTCLR_PE1 register is used to clear SEDF and DEDF in IFERSTR_PE1 and ERROVF in IFOVFSTR_PE1, and error address in IFERADR_PE1. This is write only register and read value is always “0”.

Access: IFERSTCLR_PE1 can be written only in 32/16/8-bit units.

Address: FFC6 3408_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | STCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.98 IFERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | STCLR | Error flag clear Writing 1 to this bit clears the DEDF and SEDF flags in IFERSTR_PE1; ERROVF flag in IFOVFSTR_PE1; and IFERADR_PE1. |

(10) IFOVFSTR_PE1 — IFU Data Error Count Overflow Status Register

IFOVFSTR_PE1 register monitors occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting STCLR in IFERSTCLR_PE1 register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: IFOVFSTR_PE1 can be read only in 32/16/8-bit units.

Address: FFC6 340C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.99 IFOVFSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error Overflow Flag ERROVF is set if a second error occurs while either of the error flags (DEDFF and SEDF) in the error status register is set, except when both the error address and source of the second error are the same as those of the first error. |

(11) IFERSTR_PE1 — IFU Data ECC SED/DED Status Register

IFERSTR_PE1 is the error monitor register. When all error flags are “0” and a new error occurs, error status flag is set. If data ECC 1 bit is set and the new error is data ECC 2 bit, the new error is set (does not clear the previous error flag).

IFERSTR_PE1 register is cleared by setting STCLR in IFERSTCLR_PE1 register to 1.

Access: IFERSTR_PE1 can be read only in 32/16/8-bit units.

Address: FFC6 3410_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDF | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.100 IFERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF | ECC 2-bit error Monitor Flag 0: Cleared to 0 by setting the STCLR bit to 1 in IFERSTCLR_PE1. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF was 0. |
| 0 | SEDF | ECC 1-bit error Monitor Flag 0: Cleared to 0 by setting the STCLR bit to 1 in IFERSTCLR_PE1. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF and SEDF were 0. |

(12) IFERADR_PE1 — IFU Data ECC SED/DED Address Register

IFERADR_PE1 register is used to hold the address when all error flags are not set and an error detected, or, if SEDF in IFERSTR_PE1 is set and data ECC 2-bit error is detected.

IFERADR_PE1 is cleared by setting STCLR in IFERSTCLR_PE1 register to 1.

Access: IFERADR_PE1 can be read only in 32-bit units.

Address: FFC6 3450_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | EADR[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EADR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.101 IFERADR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

(13) LSSECCCTL_PE1 — LSU Slave Data and Address ECC Control Register

LSSECCCTL_PE1 register controls the ECC error detection/correction and 1-bit error correction on access port in load store unit that is used to access local RAM from other master. Writing ECC control registers must be executed with PROT[1:0] = 01_B.

Access: LSSECCCTL_P can be read/written in 32/16-bit units.

Address: FFC6 6000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.102 LSSECCCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | Address and Data ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Address and Data ECC error detection/correction (ECCDIS = 0). 0: Correction is carried out when 1-bit is detected 1: Correction is not carried out when 1-bit is detected |
| 0 | ECCDIS | Address and Data ECC disable bit Enables or disables Address and Data ECC error detection/correction. 0: Address and Data ECC error detection/correction is enable 1: Address and Data ECC error detection/correction is disable |

(14) LSSERRINT_PE1 — LSU Slave Data and Address Error Information Control Register

LSSERRINT_PE1 register controls whether error information is reported to ECM, when data/address ECC 2-bit error and data/address ECC 1-bit error are detected.

Access: LSSERRINT_PE1 can be read/written in 32/16/8-bit units.

Address: FFC6 6004_H

Value after reset: 0000 0033_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|------------|--------|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | ADEDI E | ASEDIE | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R | R/W | R/W |

Table 31.103 LSSERRINT_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | ADEDIE | Address ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled |
| 4 | ASEDIE | Address ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | Data ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled |
| 0 | SEDIE | Data ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled |

(15) LSSERSTCLR_PE1 — LSU Slave Data and Address ECC SED/DED Status Clear Register

LSSERSTCLR_PE1 register is used to clear error flag in LSSERSTR_PE1, error overflow flag in LSSOVFSTR_PE1, and error address in LSSEADR_PE1. This is write only register and read value is always “0”.

Access: LSSERSTCLR_PE1 can be written only in 32/16/8-bit units.

Address: FFC6 6008_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | W | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.104 LSSERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | Error flag clear Writing “1” to this bit clears ADED / ASED / DED / SED in LSSERSTR_PE1, ERROVF in LSSOVFSTR_PE1, and LSSEADR_PE1. |

(16) LSSOVFSTR_PE1 — LSU Slave Data and Address Error Count Overflow Status Register

LSSOVFSTR_PE1 register monitors occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting SSTCLR in LSSERSTCLR_PE1 register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: LSSOVFSTR_PE1 can be read only in 32/16/8-bit units.

Address: FFC6 600C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.105 LSSOVFSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error overflow flag If an error flag (ADED, ASED, DED, SED in LSSERSTR_PE1) is set and a different error is detected, this bit is set. |

(17) LSSERSTR_PE1 — LSU Slave Data and Address ECC SED/DED Status Register

LSSERSTR_PE1 is the error monitor register. When all error flags are “0” and a new error occurs, error status flag is set. If address ECC 1 bit or data ECC 1 bit is set and the new error is address ECC 2 bit or data ECC 2 bit, the new error is set (does not clear the previous error flag). If multiple error causes are simultaneously detected, detected errors are all set (e.g. if DED and ADED are detected at the same access, DEDF and ADEDf are both set.). LSSERSTR_PE1 registers are cleared by setting SSTCLR in LSSERSTCLR_PE1 register to 1.

Access: LSSERSTR_PE1 can be read only in 32/16/8-bit units.

Address: FFC6 6010_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-------|-------|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | ADEDf | ASEDF | — | — | — | — | DEDf | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.106 LSSERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. |
| 7 | ADEDf | Address ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in LSSERSTCLR_PE1 Condition to “1”: ADEDf is “0” and ECC 2-bit error is detected. |
| 6 | ASEDF | Address ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in LSSERSTCLR_PE1 Condition to “1”: ASEDF is “0” and ECC 1-bit error is detected. |
| 5 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDf | Data ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in LSSERSTCLR_PE1 Condition to “1”: DEDf is “0” and ECC 2-bit error is detected. |
| 0 | SEDF | Data ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in LSSERSTCLR_PE1 Condition to “1”: SEDF is “0” and ECC 1-bit error is detected. |

(18) LSSEADR_PE1 — LSU Slave Data and Address ECC SED/DED Address Register

LSSEADR_PE1 register is used to hold the address when all error flags are not set and an error is detected, or, if ASED/SED in LSSERSTR_PE1 is set and address ECC 2 bit/data ECC 2-bit error is detected. This register stores an address that the error is corrected when address ECC 1-bit error correction is enabled. However, when address ECC 1-bit error correction is disabled or address ECC 2-bit error, an address which include ECC error is stored.

LSSEADR_PE1 is cleared by setting SSTCLR in LSSERSTCLR_PE1 register to 1.

Access: LSSEADR_PE1 can be read only in 32-bit units.

Address: FFC6 6050_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| EADR[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EADR[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.107 LSSEADR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

(19) LSMECCCTL_PE1 — LSU Master Data ECC Control Register

LSMECCCTL_PE1 register controls the ECC error detection/correction and 1-bit error correction on master port in load store unit that is used to access data in RAM or peripheral registers. Writing ECC control registers must be executed with PROT[1:0] = 01_B.

Access: LSMECCCTL_PE1 can be read/written in 32/16-bit units.

Address: FFC6 6400_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.108 LSMECCCTL_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | Data ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Data ECC error detection/correction (ECCDIS = 0). 0: Correction is done when 1-bit error is detected 1: No Correction is done when 1-bit error is detected |
| 0 | ECCDIS | Data ECC disable bit Enables or disables Data ECC error detection/correction. 0: Data ECC error detection/correction is enable 1: Data ECC error detection/correction is disable |

(20) LSMERRINT_PE1 — LSU Master Data Error Information Control Register

LSMERRINT_PE1 register controls whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected.

Access: LSMERRINT_PE1 can be read/written in 32/16/8-bit units.

Address: FFC6 6404_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.109 LSMERRINT_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | Data ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled |
| 0 | SEDIE | Data ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled |

(21) LSMERSTCLR_PE1 — LSU Master Data ECC SED/DED Status Clear Register

LSMERRSTCLR_PE1 register is used to clear error flag in LSMERSTR_PE1, error overflow flag in LSMOVFSTR_PE1, and error address in LSMEADR_PE1. This is write only register and read value is always “0”.

Access: LSMERSTCLR_PE1 can be written only in 32/16/8-bit units.

Address: FFC6 6408_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.110 LSMERSTCLR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | Error flag clear Writing “1” to this bit clears DEDF / SEDF in LSMERSTR_PE1, ERROVF in LSMOVFSTR_PE1, and LSMEADR_PE1. |

(22) LSMOVFSTR_PE1 — LSU Master Data Error Count Overflow Status Register

LSMOVFSTR_PE1 register monitors occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting SSTCLR in LSMERSTCLR_PE1 register. to 1

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: LSMOVFSTR_PE1 can be read only in 32/16/8-bit units.

Address: FFC6 640C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.111 LSMOVFSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error overflow flag If an error flag (DEDF, SEDF in LSMERSTR_PE1) is set and a different error is detected, this bit is set. |

(23) LSMERSTR_PE1 — LSU Master Data ECC SED/DED Status Register

LSMERSTR_PE1 is the error monitor register. When all error flags are “0” for a new the error occurs, error status flag is set. If data ECC 1 bit is set and the new error is data ECC 2 bit, the new error is set (does not clear the previous error flag). LSMERSTR_PE1 register is cleared by setting SSTCLR in LSMERSTCLR_PE1 register to 1.

Access: LSMERSTR_PE1 can be read only in 32/16/8-bit units.

Address: FFC6 6410_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDF | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.112 LSMERSTR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF | Data ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in LSMERSTCLR_PE1 Condition to “1”: DEDF is “0” and ECC 2-bit error is detected. |
| 0 | SEDF | Data ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in LSMERSTCLR_PE1 Condition to “1”: SEDF is “0” and ECC 1-bit error is detected. |

(24) LSMEADR_PE1 — LSU Master Data ECC SED/DED Address Register

LSMEADR_PE1 register is used to hold the address when all error flags are not set and an error is detected, or, if SEDF in LSMERSTR_PE1 is set and data ECC 2-bit error is detected.

LSMEADR_PE1 is cleared by setting SSTCLR in LSMERSTCLR_PE1 register to 1.

Access: LSMEADR_PE1 can be read only in 32-bit units.

Address: FFC6 6450_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | EADR[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EADR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.113 LSMEADR_PE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

(25) VPERRINT_SGn — System Interconnect (PBus I/F) Data Error Information Control Register (n = 0 to 2, 5)

VPERRINT_SGn (n = 0 to 2, 5) registers control whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected by ECC modules that are used only when the bit operation (RmW operation) is executed to a P-Bus peripheral register.

Access: These registers can be read/written in 32/16/8-bit units.

Address: VPERRINT_SG0: FFC6 7004_H
 VPERRINT_SG1: FFC6 7404_H
 VPERRINT_SG2: FFC6 7804_H
 VPERRINT_SG5: FFC6 8404_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.114 VPERRINT_SGn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | Data ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled |
| 0 | SEDIE | Data ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled |

(26) VPERSTCLR_SGn — System Interconnect (PBus I/F) Data ECC SED/DED Status Clear Register (n = 0 to 2, 5)

VPERSTCLR_SGn registers are used to clear error flag in VPERSTR_SGn, error overflow flag in VPOVFSTR_SGn, and error address in VPEADR_SGn. These are write only registers and read value is always “0”.

Access: These registers can be written only in 32/16/8-bit units.

Address: VPERSTCLR_SG0: FFC6 7008_H
 VPERSTCLR_SG1: FFC6 7408_H
 VPERSTCLR_SG2: FFC6 7808_H
 VPERSTCLR_SG5: FFC6 8408_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.115 VPERSTCLR_SGn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | Error flag clear Writing “1” to this bit clears DEDF / SEDF in VPERSTR_SGn, ERROVF in VPOVFSTR_SGn, and VPEADR_SGn. |

(27) VPOVFSTR_SGn — System Interconnect (PBus I/F) Data Error Count Overflow Status Register (n = 0 to 2, 5)

VPOVFSTR_SGn registers monitor occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting SSTCLR in VPERSTCLR_SGn register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: These registers can be read only in 32/16/8-bit units.

Address: VPOVFSTR_SG0: FFC6 700C_H
 VPOVFSTR_SG1: FFC6 740C_H
 VPOVFSTR_SG2: FFC6 780C_H
 VPOVFSTR_SG5: FFC6 840C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.116 VPOVFSTR_SGn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error overflow flag If an error flag (DEDF, SEDF in VPERSTR_SGn) is set and a different error is detected, this bit is set. |

(28) VPERSTR_SGn — System Interconnect (PBus I/F) Data ECC Status Register (n = 0 to 2, 5)

VPERSTR_SGn are the error monitor registers. If the error flag is “0” for a new error occurs, the error status flag is set. If data ECC 1 bit is set and the new error is data ECC 2 bit, the new error is set (does not clear the previous error flag).

VPERSTR_SGn registers are cleared by setting SSTCLR in VPERSTCLR_SGn register to 1.

Access: These registers can be read only in 32/16/8-bit units.

Address: VPERSTR_SG0: FFC6 7010_H
 VPERSTR_SG1: FFC6 7410_H
 VPERSTR_SG2: FFC6 7810_H
 VPERSTR_SG5: FFC6 8410_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDF | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.117 VPERSTR_SGn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF | Data ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in VPERSTCLR_SGn Condition to “1”: DEDF is “0” and ECC 2-bit error is detected. |
| 0 | SEDF | Data ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in VPERSTCLR_SGn Condition to “1”: SEDF is “0” and ECC 1-bit error is detected. |

(29) VPEADR_SGn — System Interconnect (PBus I/F) Data ECC SED/DED Address Register (n = 0 to 2, 5)

VPEADR_SGn registers are used to hold the address when all error flags are not set and an error is detected, or, if SEDF in VPERSTR_SGn is set and data ECC 2-bit error is detected.

VPEADR_SGn registers are cleared by setting SSTCLR in VPERSTCLR_SGn register to 1.

Access: These registers can be read only in 32-bit units.

Address: VPEADR_SG0: FFC6 7050_H
 VPEADR_SG1: FFC6 7450_H
 VPEADR_SG2: FFC6 7850_H
 VPEADR_SG5: FFC6 8450_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| EADR[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EADR[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.118 VPEADR_SGn Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

(30) VCECCCTL_PDMA — System Interconnect Data ECC Control Register

VCECCCTL_PDMA register controls the ECC error detection/correction and 1-bit error correction on master port of DMA that is used to read to a source from DMA. Writing ECC control registers must be executed with $PROT[1:0] = 01_B$.

Access: VCECCCTL_PDMA can be read/written in 32/16-bit units.

Address: FFC6 A000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | — | — | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.119 VCECCCTL_PDMA Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | SECDIS | Data ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Data ECC error detection/correction (ECCDIS = 0). 0: Correction is done when 1-bit error is detected 1: No Correction is done when 1-bit error is detected |
| 0 | ECCDIS | Data ECC disable bit Enables or disables Data ECC error detection/correction. 0: Data ECC error detection/correction is enable 1: Data ECC error detection/correction is disable |

(31) VCERRINT_PDMA — System Interconnect Error Information Control Register

VCERRINT_PDMA register controls whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected.

Access: VCERRINT_PDMA can be read/written in 32/16/8-bit units.

Address: FFC6 A004_H

Value after reset: 0000 0003_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.120 VCERRINT_PDMA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | Data ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled |
| 0 | SEDIE | Data ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled |

(32) VCERSTCLR_PDMA — System Interconnect Data ECC SED/DED Status Clear Register

VCERSTCLR_PDMA register is used to clear error flag in VCERSTR_PDMA, error overflow flag in VCOVFSTR_PDMA, and error address in VCEADR_PDMA. This is write only register and read value is always “0”.

Access: VCERSTCLR_PDMA can be written only in 32/16/8-bit units.

Address: FFC6 A008_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.121 VCERSTCLR_PDMA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | Error flag clear Writing “1” to this bit clears DEDF / SEDF in VCERSTR_PDMA, ERROVF in VCOVFSTR_PDMA, and VCEADR_PDMA. |

(33) VCOVFSTR_PDMA — System Interconnect Data Error Count Overflow Status Register

VCOVFSTR_PDMA register monitors occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting SSTCLR in VCERSTCLR_PDMA register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: VCOVFSTR_PDMA can be read only in 32/16/8-bit units.

Address: FFC6 A00C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.122 VCOVFSTR_PDMA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error overflow flag If an error flag (DEDF, SEDF in VCERSTR_PDMA) is set and a different error is detected, this bit is set. |

(34) VCERSTR_PDMA — System Interconnect Data ECC SED/DED Status Register

VCERSTR_PDMA is the error monitor register. If the error flag is “0” and a new error occurs, the error status flag is set. If data ECC 1-bit error monitor flag is set and the new error is data ECC 2-bit error, the new error is set (does not clear the previous error flag). VCERSTR_PDMA register is cleared by setting SSTCLR in VCERSTCLR_PDMA register to 1.

Access: VCERSTR_PDMA can be read only in 32/16/8-bit units.

Address: FFC6 A010_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DEDF | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.123 VCERSTR_PDMA Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF | Data ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in VCERSTCLR_PDMA Condition to “1”: DEDF is “0” and ECC 2-bit error is detected. |
| 0 | SEDF | Data ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in VCERSTCLR_PDMA Condition to “1”: SEDF is “0” and ECC 1-bit error is detected. |

(35) VCEADR_PDMA — System Interconnect Data ECC SED/DED Address Register

VCEADR_PDMA register is used to hold the address when all error flags are not set and an error is detected, or, if SEDF in VCERSTR_PDMA is set and data ECC 2-bit error is detected.

VCEADR_PDMA is cleared by setting SSTCLR in VCERSTCLR_PDMA register to 1.

Access: VCEADR_PDMA can be read only in 32-bit units.

Address: FFC6 A050_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| EADR[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EADR[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.124 VCEADR_PDMA Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

(36) APECCCTL_PFSS — P-bus Data and Address ECC Control Register

APECCCTL_PFSS register controls the ECC error detection/correction and 1-bit error correction on peripheral IP group 0. Writing ECC control registers must be executed with PROT[1:0] = 01_B.

Access: APECCCTL_PFSS can be read/written in 32/16-bit units.

Address: FFC6 C000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|---------|---------|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | ASECDIS | AECDDIS | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |

Table 31.125 APECCCTL_PFSS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | ASECDIS | Address ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Address ECC error detection/correction (AECDDIS = 0). 0: Correction is carried out when 1-bit is detected 1: Correction is not carried out when 1-bit is detected |
| 2 | AECDDIS | Address ECC disable bit Enables or disables Address ECC error detection/correction. 0: Address ECC error detection/correction is enable 1: Address ECC error detection/correction is disable |
| 1 | SECDIS | Data ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Data ECC error detection/correction (ECCDIS = 0). 0: Correction is carried out when 1-bit is detected 1: Correction is not carried out when 1-bit is detected |
| 0 | ECCDIS | Data ECC disable bit Enables or disables Data ECC error detection/correction. 0:Data ECC error detection/correction is enable 1:Data ECC error detection/correction is disable |

(37) APERRINT_PFSS — P-bus Error Information Control Register

APERRINT_PFSS register controls whether error information is reported to ECM, when data/address ECC 2-bit error and data/address ECC 1-bit error are detected.

Access: APERRINT_PFSS can be read/written in 32/16/8-bit units.

Address: FFC6 C004_H

Value after reset: 0000 0073_H

| | | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|------------|--------|----|----|-------|-------|---|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | — | — | — | — | — | — | — | — | — | — | ADEDI E | ASEDIE | — | — | DEDIE | SEDIE | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R | R/W | R/W | |

Table 31.126 APERRINT_PFSS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | ADEDIE | Address ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled |
| 4 | ASEDIE | Address ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | Data ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled |
| 0 | SEDIE | Data ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled |

(38) APERSTCLR_PFSS — P-bus Data and Address ECC SED/DED Status Clear Register

APERSTCLR_PFSS register is used to clear error flag in APERSTR_PFSS, error overflow flag in APOVFSTR_PFSS, and error address in APEADR_PFSS. This is write only register and read value is always “0”.

Access: APERSTCLR_PFSS can be written only in 32/16/8-bit units.

Address: FFC6 C008_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.127 APERSTCLR_PFSS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | Error flag clear Writing “1” to this bit clears ADED / ASED / DED / SED in APERSTR_PFSS, ERROVF in APOVFSTR_PFSS, and APEADR_PFSS. |

(39) APOVFSTR_PFSS — P-bus Data and Address Error Count Overflow Status Register

APOVFSTR_PFSS register monitors occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting SSTCLR in APERSTCLR_PFSS register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: APOVFSTR_PFSS can be read only in 32/16/8-bit units.

Address: FFC6 C00C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.128 APOVFSTR_PFSS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error overflow flag If an error flag (ADED, ASED, DED, SED in APERSTR_PFSS) is set and a different error is detected, this bit is set. |

(40) APERSTR_PFSS — P-bus Data and Address ECC SED/DED Status Register

APERSTR_PFSS is the error monitor register. If the error flag is “0” for a new error occurs, the error status flag is set. If address ECC 1 bit or data ECC 1 bit is set and the new error is address ECC 2 bit or data ECC 2 bit, the new error is set (does not clear the previous error flag). If multiple error causes are simultaneously detected, detected errors are all set (e.g. if DED and ADED are detected at the same access, DEDF and ADEDF are both set.). APERSTR_PFSS register is cleared by setting SSTCLR in APERSTCLR_PFSS register to 1.

Access: APERSTR_PFSS can be read only in 32/16/8-bit units.

Address: FFC6 C010_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-------|-------|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | ADEDF | ASEDF | — | — | — | — | DEDF | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.129 APERSTR_PFSS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. |
| 7 | ADEDF | Address ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APERSTCLR_PFSS Condition to “1”: ADEDF is “0” and ECC 2-bit error is detected. |
| 6 | ASEDF | Address ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APERSTCLR_PFSS Condition to “1”: ASEDF is “0” and ECC 1-bit error is detected. |
| 5 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DEDF | Data ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APERSTCLR_PFSS Condition to “1”: DEDF is “0” and ECC 2-bit error is detected. |
| 0 | SEDF | Data ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APERSTCLR_PFSS Condition to “1”: SEDF is “0” and ECC 1-bit error is detected. |

(41) APEADR_PFSS — P-bus Data and Address ECC SED/DED Address Register

APEADR_PFSS register is used to hold the address when all error flags are not set and an error is detected, or, if ASEDf/SEDF in APERSTR_PFSS is set and address ECC 2 bit/data ECC 2-bit error is detected. This register stores an address that the error is corrected when address ECC 1-bit error correction is enabled. However, when address ECC 1-bit error correction is disabled or address ECC 2-bit error, an address which include ECC error is stored.

APEADR_PFSS is cleared by setting SSTCLR in APERSTCLR_PFSS register to 1.

Access: APEADR_PFSS can be read only in 32-bit units.

Address: FFC6 C050_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| EADR[31:16] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EADR[15:0] | | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.130 APEADR_PFSS Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

(42) APECnECCCTL — P-bus Data and Address ECC Control Register

APECnECCCTL register controls the ECC error detection/correction and 1-bit error correction on peripheral IP group 1, 2 or 5 (refer to **Table 31.132**). Writing ECC control registers must be executed with $PROT[1:0] = 01_B$.

Access: These registers can be read/written in 32/16-bit units.

Address: APEC0ECCCTL: FFDD D800_H
 APEC1ECCCTL: FFF9 4800_H
 APEC2ECCCTL: FFC4 0800_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------|-------|----|----|----|----|----|----|----|----|----|----|---------|---------|--------|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PROT1 | PROT0 | — | — | — | — | — | — | — | — | — | — | ASECDIS | AECCDIS | SECDIS | ECCDIS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |

Table 31.131 APECnECCCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 16 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 15, 14 | PROT[1:0] | Protection bit 01: write is enable Other: write is disable These bits are always read as 0. |
| 13 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 | ASECDIS | Address ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Address ECC error detection/correction (AECCDIS = 0). 0: Correction is carried out when 1-bit is detected 1: Correction is not carried out when 1-bit is detected |
| 2 | AECCDIS | Address ECC disable bit Enables or disables Address ECC error detection/correction. 0: Address ECC error detection/correction is enable 1: Address ECC error detection/correction is disable |
| 1 | SECDIS | Data ECC 1-bit error correction enable bit Enables or disables 1-bit error correction when using Data ECC error detection/correction (ECCDIS = 0). 0: Correction is carried out when 1-bit is detected 1: Correction is not carried out when 1-bit is detected |
| 0 | ECCDIS | Data ECC disable bit Enables or disables Data ECC error detection/correction. 0: Data ECC error detection/correction is enable 1: Data ECC error detection/correction is disable |

Table 31.132 Target Peripherals of APECnxxxx

| ECC module | target peripheral |
|------------|-----------------------|
| APEC0xxxx | Peripheral IP group 1 |
| APEC1xxxx | Peripheral IP group 2 |
| APEC2xxxx | Peripheral IP group 5 |

(43) APECnERRINT — P-bus Error Information Control Register

APECnERRINT registers control whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

Access: These registers can be read/written in 32/16/8-bit units.

Address: APEC0ERRINT: FFDD D804_H
 APEC1ERRINT: FFF9 4804_H
 APEC2ERRINT: FFC4 0804_H

Value after reset: 0000 0073_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|------------|--------|----|----|-------|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | ADEDI E | ASEDIE | — | — | DEDIE | SEDIE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R | R/W | R/W |

Table 31.133 APECnERRINT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5 | ADEDIE | Address ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled |
| 4 | ASEDIE | Address ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Address ECC error detection/correction is enabled 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | DEDIE | Data ECC 2-bit error report enable bit Controls error report of 2-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled |
| 0 | SEDIE | Data ECC 1-bit error report enable bit Controls error report of 1-bit error detection when Data ECC error detection/correction is enabled 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled |

(44) APECnSTCLR — P-bus Data and Address ECC SED/DED Status Clear Register

APECnSTCLR registers are used to clear error flag in APECn1STERSTR, error overflow flag in APECnOVFSTR, and error address in APECn1STEADR0. These are write only registers and read value is always “0”.

Access: These registers can be written only in 32/16/8-bit units.

Address: APEC0STCLR: FFDD D808_H
 APEC1STCLR: FFF9 4808_H
 APEC2STCLR: FFC4 0808_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSTCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W |

Table 31.134 APECnSTCLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When writing, write the value after reset. |
| 0 | SSTCLR | Error flag clear Writing “1” to this bit clears ADED _F / ASED _F / DED _F / SED _F in APECn1STERSTR, ERROVF in APECnOVFSTR, and APECn1STEADR0. |

(45) APECnOVFSTR — P-bus Data and Address Error Count Overflow Status Register

APECnOVFSTR registers monitor occurrence of error overflow. Overflow occurs when different error*¹ is detected when the error status has been already set. If the error in error status is the same error (same error cause and same error address) with the errors already stored, this flag is not set. ERROVF flag is cleared by setting SSTCLR in APECnSTCLR register to 1.

Note 1. Different means that the error satisfying one of the following conditions.

- Detected address is different
- Detected error status flag is different.

Access: These registers can be read only in 32/16/8-bit units.

Address: APEC0OVFSTR: FFDD D80C_H
APEC1OVFSTR: FFF9 480C_H
APEC2OVFSTR: FFC4 080C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ERROVF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.135 APECnOVFSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | ERROVF | Error overflow flag If an error flag (ADED, ASED, DED, SED in APECn1STERSTR) is set and a different error is detected, this bit is set. |

(46) APECn1STERSTR — P-bus Data and Address ECC SED/DED Status Register

APECn1STERSTR are the error monitor registers. If the error flag is “0” for a new error occurs, the error status flag is set. If address ECC 1-bit error monitor flag or data ECC 1-bit error monitor flag is set and the new error is address ECC 2-bit error or data ECC 2-bit error, the new error is set (does not clear the previous error flag). If multiple error causes are simultaneously detected, detected errors are all set (e.g. if DED and ADED are detected at the same access, DEDF and ADED are both set.).

APECn1STERSTR registers are cleared by setting SSTCLR in APECnSTCLR register to 1.

Access: These registers can be read only in 32/16/8-bit units.

Address: APEC01STERSTR: FFDD D810_H
 APEC11STERSTR: FFF9 4810_H
 APEC21STERSTR: FFC4 0810_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|------|-------|----|----|----|----|-----|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | ADED | ASEDF | — | — | — | — | DED | SEDF |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.136 APECn1STERSTR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 8 | Reserved | When read, the value after reset is read. |
| 7 | ADED | Address ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APECnSTCLR Condition to “1”: ADED is “0” and ECC 2-bit error is detected. |
| 6 | ASEDF | Address ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APECnSTCLR Condition to “1”: ASEDF is “0” and ECC 1-bit error is detected. |
| 5 to 2 | Reserved | When read, the value after reset is read. |
| 1 | DED | Data ECC 2-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APECnSTCLR Condition to “1”: DED is “0” and ECC 2-bit error is detected. |
| 0 | SEDF | Data ECC 1-bit error monitor flag Condition to “0”: write “1” to SSTCLR in APECnSTCLR Condition to “1”: SEDF is “0” and ECC 1-bit error is detected. |

(47) APECn1STEADR0 — P-bus Data and Address ECC SED/DED Address Register

APECn1STEADR0 registers are used to hold the address when all error flags are not set and an error is detected, or, if ASEDf/SEDF in APECn1STERSTR is set and address ECC 2 bit/data ECC 2-bit error is detected. This register stores an address that the error is corrected when address ECC 1-bit error correction is enabled. However, when address ECC 1-bit error correction is disabled or address ECC 2-bit error, an address which include ECC error is stored.

APECn1STEADR0 registers are cleared by setting SSTCLR in APECnSTCLR register to 1.

Access: These registers can be read only in 32-bit units.

Address: APEC01STEADR0: FFDD D850_H
 APEC11STEADR0: FFF9 4850_H
 APEC21STEADR0: FFC4 0850_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | EADR[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | EADR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.137 APECn1STEADR0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|-------------------------|
| 31 to 0 | EADR[31:0] | Error detection address |

31.2.9.6 Test Function

The self-diagnosis operation varies depending on the target ECC decoder.

Error injection is not supported for VPxxxx_SGn that is used only when the bit operation is executed.

(1) Self-diagnosis of the Data ECC check function (Slave)

- (a) Prepare data with an error to check a data ECC decoder on the bus in global RAM (refer to **Section 31.2.5.4** about how to write an error data into global RAM).
- (b) Set ECCDIS bit in GRECCCTL_VCI2GRAM to 1 to transfer data with error into ECC decoder on the bus. Also, set ECCDIS bit in VCECCCTL_PDMA to 1 to pass through DMA ECC decoder on the bus if DMA ECC decoder is not a target of self-diagnosis.
- (c) For a DMA channel, specify the address of error data in global RAM to the source address register and the address of area covered by target ECC decoder to the destination address register, respectively (for details on channel setup, refer to **Section 7, DMA**).

After that, self-diagnosis of the data ECC decoder is possible by setting software DMA transfer request flag of the channel.

This operation can be used for the following ECC decoders:

- LSU Slave data ECC: LSSxxxx of module BECCPE1 (local RAM)
- P-bus data ECC: module BECCPBAm (a peripheral IP on each P-bus group)
- DMA Data ECC on Bus: xxxx_PDMA of module BECCSIC

The notes in () indicate the area specified as the destination.

(2) Self-diagnosis of the Data ECC check function (Master)

- (a) Prepare data with an error to check a data ECC decoder on the bus in global RAM (refer to **Section 31.2.5.4** about how to write an error data into global RAM).
- (b) Set ECCDIS bit in GRECCCTL_GRAMC to transfer data with error into ECC decoder on the bus.

After that, self-diagnosis of the data ECC decoder in instruction fetch unit is possible by fetching instruction from global RAM.

This operation can be used for the following ECC decoders:

- LSU Master data ECC: LSMxxxx of module BECCPE1
- IFU data ECC: IFxxxx of module BECCPE1

(3) Self-diagnosis of the Address ECC check function

- (a) Set the ECCTST bit in the address ECC test control register (refer to **Section 7.8.2.20**) to 1 to enter test mode.
- (b) Set the RWSEL bit in the address ECC test control register to select the DMA cycle to inject an error into address ECC bit.
- (c) Set the ECCDAT[6:0] in the address ECC test data register (refer to **Section 7.8.2.21**) to an error data.

(d) For a DMA channel

If RWSEL = 0, specify the address of area covered by target ECC decoder in the source address register. The destination address can be arbitrarily defined. If RWSEL = 1, specify the address of area covered by target ECC decoder in the destination address register. The source address can be arbitrarily defined.

After that, self-diagnosis of the address ECC decoder is possible by setting software DMA transfer request flag of the channel.

This operation can be used for the following ECC decoders:

- LSU Slave address ECC: LSSxxxx of module BECCPE1 (local RAM)
- P-bus address ECC: module BECCPBAm (a peripheral on each P-bus group)
- global RAM address ECC: xxxx_GRAMC in module ECCGRC (global RAM)
- address ECC for the bus width conversion in Flash interface: module BECCFLI (Code Flash)

The notes in () indicate the area specified as the destination or source.

31.2.10 ECC Function

31.2.10.1 ECC code for RAM Modules exception instruction cache RAM (data)

(1) Generating an Error Correcting Code (ECC)

The following formulas are used to generate 7-bit error correction code (ecc6 to ecc2, ecc1z and ecc0z) for 32-bit data (d32 to d0). The lowest 2 bits of error correction code are inverted. With this measure, all 0s and all 1s of data and error correction code become an illegal combination.

- $\text{ecc6} = (\text{d13} \wedge \text{d12} \wedge \text{d11} \wedge \text{d10} \wedge \text{d9} \wedge \text{d8} \wedge \text{d7} \wedge \text{d6} \wedge \text{d5} \wedge \text{d4} \wedge \text{d3} \wedge \text{d2} \wedge \text{d1} \wedge \text{d0})$
- $\text{ecc5} = (\text{d23} \wedge \text{d22} \wedge \text{d21} \wedge \text{d20} \wedge \text{d19} \wedge \text{d18} \wedge \text{d17} \wedge \text{d16} \wedge \text{d15} \wedge \text{d14} \wedge \text{d3} \wedge \text{d2} \wedge \text{d1} \wedge \text{d0})$
- $\text{ecc4} = (\text{d29} \wedge \text{d28} \wedge \text{d27} \wedge \text{d26} \wedge \text{d25} \wedge \text{d24} \wedge \text{d17} \wedge \text{d16} \wedge \text{d15} \wedge \text{d14} \wedge \text{d7} \wedge \text{d6} \wedge \text{d5} \wedge \text{d4})$
- $\text{ecc3} = (\text{d31} \wedge \text{d30} \wedge \text{d26} \wedge \text{d25} \wedge \text{d24} \wedge \text{d20} \wedge \text{d19} \wedge \text{d18} \wedge \text{d14} \wedge \text{d10} \wedge \text{d9} \wedge \text{d8} \wedge \text{d4} \wedge \text{d0})$
- $\text{ecc2} = (\text{d31} \wedge \text{d30} \wedge \text{d28} \wedge \text{d27} \wedge \text{d24} \wedge \text{d22} \wedge \text{d21} \wedge \text{d18} \wedge \text{d15} \wedge \text{d12} \wedge \text{d11} \wedge \text{d8} \wedge \text{d5} \wedge \text{d1})$
- $\text{ecc1z} = \text{!}(\text{d30} \wedge \text{d29} \wedge \text{d27} \wedge \text{d25} \wedge \text{d23} \wedge \text{d21} \wedge \text{d19} \wedge \text{d16} \wedge \text{d13} \wedge \text{d11} \wedge \text{d9} \wedge \text{d6} \wedge \text{d2} \wedge \text{d0})$
- $\text{ecc0z} = \text{!}(\text{d31} \wedge \text{d29} \wedge \text{d28} \wedge \text{d26} \wedge \text{d23} \wedge \text{d22} \wedge \text{d20} \wedge \text{d17} \wedge \text{d13} \wedge \text{d12} \wedge \text{d10} \wedge \text{d7} \wedge \text{d3} \wedge \text{d0})$

Where \wedge denotes an Exclusive OR.

(2) Error Detection

According to the following formula, 7-bit syndromes (synd6 to synd0) are generated through 32-bit data (d31 to d0) and 7-bit error correction code (ecc6 to ecc2, ecc1z and ecc0z) which are read from the RAM.

$$\begin{pmatrix} 1000000 & 00000000 & 00000000 & 00111111 & 11111111 \\ 0100000 & 00000000 & 11111111 & 11000000 & 00001111 \\ 0010000 & 00111111 & 00000011 & 11000000 & 11110000 \\ 0001000 & 11000111 & 00011100 & 01000111 & 00010001 \\ 0000100 & 11011001 & 01100100 & 10011001 & 00100010 \\ 0000010 & 01101010 & 10101001 & 00101010 & 01000101 \\ 0000001 & 10110100 & 11010010 & 00110100 & 10001001 \end{pmatrix} \times \begin{pmatrix} \text{ecc6} \\ \text{ecc5} \\ \vdots \\ \text{!ecc1z} \\ \text{!ecc0z} \\ \text{d31} \\ \text{d30} \\ \vdots \\ \text{d1} \\ \text{d0} \end{pmatrix} = \begin{pmatrix} \text{synd6} \\ \text{synd5} \\ \vdots \\ \text{synd4} \\ \text{synd3} \\ \text{synd2} \\ \text{synd1} \\ \text{synd0} \end{pmatrix}$$

Figure 31.1 Matrix for ECC Decoding

When synd6 to synd0 are all 0s, error detection is not performed. When some of synd6 to synd0 are 1, the target bit for correction is identified according to **Table 31.138** and the error bit is corrected. When synd6 to synd0 have values not indicated in **Table 31.138**, 2-bit error is detected.

Table 31.138 Correspondence Between Correction Target and Syndrome (1/2)

| synd[6:0] | Error Bit Position | synd[6:0] | Error Bit Position |
|-----------|--------------------|-----------|--------------------|
| 1000000 | ECC data bit 6 | 0101100 | RAM data bit 18 |
| 0100000 | ECC data bit 5 | 0110001 | RAM data bit 17 |
| 0010000 | ECC data bit 4 | 0110010 | RAM data bit 16 |
| 0001000 | ECC data bit 3 | 0110100 | RAM data bit 15 |
| 0000100 | ECC data bit 2 | 0111000 | RAM data bit 14 |
| 0000010 | ECC data bit 1 | 1000011 | RAM data bit 13 |
| 0000001 | ECC data bit 0 | 1000101 | RAM data bit 12 |

Table 31.138 Correspondence Between Correction Target and Syndrome (2/2)

| synd[6:0] | Error Bit Position | synd[6:0] | Error Bit Position |
|------------------|---------------------------|------------------|---------------------------|
| 0001101 | RAM data bit 31 | 1000110 | RAM data bit 11 |
| 0001110 | RAM data bit 30 | 1001001 | RAM data bit 10 |
| 0010011 | RAM data bit 29 | 1001010 | RAM data bit 9 |
| 0010101 | RAM data bit 28 | 1001100 | RAM data bit 8 |
| 0010110 | RAM data bit 27 | 1010001 | RAM data bit 7 |
| 0011001 | RAM data bit 26 | 1010010 | RAM data bit 6 |
| 0011010 | RAM data bit 25 | 1010100 | RAM data bit 5 |
| 0011100 | RAM data bit 24 | 1011000 | RAM data bit 4 |
| 0100011 | RAM data bit 23 | 1100001 | RAM data bit 3 |
| 0100101 | RAM data bit 22 | 1100010 | RAM data bit 2 |
| 0100110 | RAM data bit 21 | 1100100 | RAM data bit 1 |
| 0101001 | RAM data bit 20 | 1101011 | RAM data bit 0 |
| 0101010 | RAM data bit 19 | 0000000 | No error |

31.2.10.2 EDC Function for Instruction cache RAM (data)

(1) Generating an Error Detecting Code (EDC)

The following formulas are used to generate 8-bit error detecting codes (ecc7 to ecc2, ecc1z and ecc0z) for 64-bit data (d63 to d0). The lowest 2 bits of error detection code are inverted. With this measure, all 0s and all 1s of data and error correction code become an illegal combination.

- $ecc7 = (d63 \wedge d62 \wedge d61 \wedge d60 \wedge d48 \wedge d31 \wedge d30 \wedge d29 \wedge d28 \wedge d27 \wedge d26 \wedge d25 \wedge d24 \wedge d21 \wedge d20 \wedge d19 \wedge d18 \wedge d14 \wedge d13 \wedge d10 \wedge d9 \wedge d7 \wedge d5 \wedge d3 \wedge d1 \wedge d0)$
- $ecc6 = (d63 \wedge d61 \wedge d59 \wedge d57 \wedge d56 \wedge d55 \wedge d54 \wedge d53 \wedge d52 \wedge d40 \wedge d23 \wedge d22 \wedge d21 \wedge d20 \wedge d19 \wedge d18 \wedge d17 \wedge d16 \wedge d13 \wedge d12 \wedge d11 \wedge d10 \wedge d6 \wedge d5 \wedge d2 \wedge d1)$
- $ecc5 = (d62 \wedge d61 \wedge d58 \wedge d57 \wedge d55 \wedge d53 \wedge d51 \wedge d49 \wedge d48 \wedge d47 \wedge d46 \wedge d45 \wedge d44 \wedge d32 \wedge d15 \wedge d14 \wedge d13 \wedge d12 \wedge d11 \wedge d10 \wedge d9 \wedge d8 \wedge d5 \wedge d4 \wedge d3 \wedge d2)$
- $ecc4 = (d61 \wedge d60 \wedge d59 \wedge d58 \wedge d54 \wedge d53 \wedge d50 \wedge d49 \wedge d47 \wedge d45 \wedge d43 \wedge d41 \wedge d40 \wedge d39 \wedge d38 \wedge d37 \wedge d36 \wedge d24 \wedge d7 \wedge d6 \wedge d5 \wedge d4 \wedge d3 \wedge d2 \wedge d1 \wedge d0)$
- $ecc3 = (d63 \wedge d62 \wedge d61 \wedge d60 \wedge d59 \wedge d58 \wedge d57 \wedge d56 \wedge d53 \wedge d52 \wedge d51 \wedge d50 \wedge d46 \wedge d45 \wedge d42 \wedge d41 \wedge d39 \wedge d37 \wedge d35 \wedge d33 \wedge d32 \wedge d31 \wedge d30 \wedge d29 \wedge d28 \wedge d16)$
- $ecc2 = (d55 \wedge d54 \wedge d53 \wedge d52 \wedge d51 \wedge d50 \wedge d49 \wedge d48 \wedge d45 \wedge d44 \wedge d43 \wedge d42 \wedge d38 \wedge d37 \wedge d34 \wedge d33 \wedge d31 \wedge d29 \wedge d27 \wedge d25 \wedge d24 \wedge d23 \wedge d22 \wedge d21 \wedge d20 \wedge d8)$
- $ecc1z = \!(d47 \wedge d46 \wedge d45 \wedge d44 \wedge d43 \wedge d42 \wedge d41 \wedge d40 \wedge d37 \wedge d36 \wedge d35 \wedge d34 \wedge d30 \wedge d29 \wedge d26 \wedge d25 \wedge d23 \wedge d21 \wedge d19 \wedge d17 \wedge d16 \wedge d15 \wedge d14 \wedge d13 \wedge d12 \wedge d0)$
- $ecc0z = \!(d56 \wedge d39 \wedge d38 \wedge d37 \wedge d36 \wedge d35 \wedge d34 \wedge d33 \wedge d32 \wedge d29 \wedge d28 \wedge d27 \wedge d26 \wedge d22 \wedge d21 \wedge d18 \wedge d17 \wedge d15 \wedge d13 \wedge d11 \wedge d9 \wedge d8 \wedge d7 \wedge d6 \wedge d5 \wedge d4)$

Where \wedge denotes an Exclusive OR.

(2) Error Detection

According to the following formula, 8-bit syndromes (synd7 to synd0) are generated through 64-bit data (d63 to d0) and 8-bit error detecting code (ecc7 to ecc2, ecc1z and ecc0z) which are read from the RAM.

$$\begin{pmatrix}
 10000000 & 11110000 & 00000001 & 00000000 & 00000000 & 11111111 & 00111100 & 01100110 & 10101011 \\
 01000000 & 10101011 & 11110000 & 00000001 & 00000000 & 00000000 & 11111111 & 00111100 & 01100110 \\
 00100000 & 01100110 & 10101011 & 11110000 & 00000001 & 00000000 & 00000000 & 11111111 & 00111100 \\
 00010000 & 00111100 & 01100110 & 10101011 & 11110000 & 00000001 & 00000000 & 00000000 & 11111111 \\
 00001000 & 11111111 & 00111100 & 01100110 & 10101011 & 11110000 & 00000001 & 00000000 & 00000000
 \end{pmatrix}
 \times
 \begin{pmatrix}
 ecc7 \\
 ecc6 \\
 : \\
 !ecc1z \\
 !ecc0z \\
 d31 \\
 d30 \\
 : \\
 d1 \\
 d0
 \end{pmatrix}
 =
 \begin{pmatrix}
 synd7 \\
 synd6 \\
 synd5 \\
 synd4 \\
 synd3 \\
 synd2 \\
 synd1 \\
 synd0
 \end{pmatrix}$$

Figure 31.2 Matrix for EDC Decoding

When synd7 to synd0 are all 0s, error detection is not performed. When some of synd7 to synd0 are 1, the target bit for detection is identified according to **Table 31.139**. When synd7 to synd0 have values not indicated in **Table 31.139**, 2-bit error is detected.

Table 31.139 Syndrome Codes and the Respective Error Bit Position

| synd[7:0] | Error Bit Position | synd[7:0] | Error Bit Position | synd[7:0] | Error Bit Position |
|-----------|--------------------|-----------|--------------------|-----------|--------------------|
| 10000000 | ECC data bit 7 | 00101010 | RAM data bit 46 | 11000111 | RAM data bit 21 |
| 01000000 | ECC data bit 6 | 00111110 | RAM data bit 45 | 11000100 | RAM data bit 20 |
| 00100000 | ECC data bit 5 | 00100110 | RAM data bit 44 | 11000010 | RAM data bit 19 |
| 00010000 | ECC data bit 4 | 00010110 | RAM data bit 43 | 11000001 | RAM data bit 18 |
| 00001000 | ECC data bit 3 | 00001110 | RAM data bit 42 | 01000011 | RAM data bit 17 |
| 00000100 | ECC data bit 2 | 00011010 | RAM data bit 41 | 01001010 | RAM data bit 16 |
| 00000010 | ECC data bit 1 | 01010010 | RAM data bit 40 | 00100011 | RAM data bit 15 |
| 00000001 | ECC data bit 0 | 00011001 | RAM data bit 39 | 10100010 | RAM data bit 14 |
| 11001000 | RAM data bit 63 | 00010101 | RAM data bit 38 | 11100011 | RAM data bit 13 |
| 10101000 | RAM data bit 62 | 00011111 | RAM data bit 37 | 01100010 | RAM data bit 12 |
| 11111000 | RAM data bit 61 | 00010011 | RAM data bit 36 | 01100001 | RAM data bit 11 |
| 10011000 | RAM data bit 60 | 00001011 | RAM data bit 35 | 11100000 | RAM data bit 10 |
| 01011000 | RAM data bit 59 | 00000111 | RAM data bit 34 | 10100001 | RAM data bit 9 |
| 00111000 | RAM data bit 58 | 00001101 | RAM data bit 33 | 00100101 | RAM data bit 8 |
| 01101000 | RAM data bit 57 | 00101001 | RAM data bit 32 | 10010001 | RAM data bit 7 |
| 01001001 | RAM data bit 56 | 10001100 | RAM data bit 31 | 01010001 | RAM data bit 6 |
| 01100100 | RAM data bit 55 | 10001010 | RAM data bit 30 | 11110001 | RAM data bit 5 |
| 01010100 | RAM data bit 54 | 10001111 | RAM data bit 29 | 00110001 | RAM data bit 4 |
| 01111100 | RAM data bit 53 | 10001001 | RAM data bit 28 | 10110000 | RAM data bit 3 |
| 01001100 | RAM data bit 52 | 10000101 | RAM data bit 27 | 01110000 | RAM data bit 2 |
| 00101100 | RAM data bit 51 | 10000011 | RAM data bit 26 | 11010000 | RAM data bit 1 |
| 00011100 | RAM data bit 50 | 10000110 | RAM data bit 25 | 10010010 | RAM data bit 0 |
| 00110100 | RAM data bit 49 | 10010100 | RAM data bit 24 | 00000000 | No ECC error |
| 10100100 | RAM data bit 48 | 01000110 | RAM data bit 23 | | |
| 00110010 | RAM data bit 47 | 01000101 | RAM data bit 22 | | |

31.2.10.3 ECC Function for Code Flash

(1) Generating an Error Correcting Code (ECC)

The following formulas are used to generate 9-bit error detecting codes (ecc8z, ecc7, ecc6z, ecc5, ecc4, ecc3z, ecc2, ecc1z and ecc0z) for 128-bit data (d127 to d0). 5-bits (ecc8, ecc6, ecc3, ecc1 and ecc0) of error correction code are inverted. With this measure, all 0s and all 1s of data and error correction code become an illegal combination.

$$\begin{pmatrix}
 10000010 & 10000100 & 10001111 & 11111111 & 11111111 & 11000000 & 00000000 & 00000000 & 00000000 & 00000111 & 10111011 & 01001000 & 01000100 & 10110010 & 00111110 & 01011101 \\
 01000001 & 01000010 & 10000111 & 11100000 & 00000000 & 00111111 & 11111111 & 10000000 & 00000000 & 00000111 & 01110110 & 10101000 & 10001001 & 01010011 & 00011111 & 00101111 \\
 10100000 & 00100001 & 01000100 & 00011111 & 00000000 & 00111110 & 00000000 & 01111111 & 11100000 & 00000110 & 11101101 & 10011001 & 00010010 & 01100011 & 10001111 & 10010111 \\
 01010000 & 10010000 & 01000010 & 00010000 & 11110000 & 00100001 & 11100000 & 01110000 & 00011111 & 10000101 & 11011100 & 01110010 & 00100011 & 10001111 & 11000111 & 11001010 \\
 00101000 & 01001000 & 00100001 & 00001000 & 10001110 & 00010001 & 00011100 & 01000111 & 00011100 & 01110011 & 11000011 & 11110100 & 00111100 & 00001111 & 11100001 & 11100101 \\
 00010100 & 00100100 & 00100000 & 10000100 & 01001001 & 10001000 & 10010011 & 00100100 & 11010011 & 01101000 & 00111111 & 11110111 & 11000000 & 00001101 & 11110010 & 11110010 \\
 00001010 & 00010010 & 00010000 & 01000010 & 00100101 & 01000100 & 01001010 & 10010010 & 10101010 & 11011000 & 00000000 & 00001111 & 11111111 & 11110100 & 11111001 & 01111001 \\
 00000101 & 00001001 & 00011000 & 00100001 & 00010010 & 11000010 & 00100101 & 10001001 & 01100101 & 10111000 & 00000000 & 00000111 & 11111111 & 11111000 & 01111100 & 10111110 \\
 11111111 & 11111111 & 11111100 & 00000000 & 00000000 & 00000000 & 00000000 & 00000000 & 00000000 & 00000111 & 11111111 & 11111111 & 11111111 & 11111100 & 00000000 & 00000000
 \end{pmatrix}
 \times
 \begin{pmatrix}
 d127 \\
 d126 \\
 : \\
 : \\
 d1 \\
 d0
 \end{pmatrix}
 =
 \begin{pmatrix}
 ecc8 \\
 ecc7 \\
 ecc6 \\
 ecc5 \\
 ecc4 \\
 ecc3 \\
 ecc2 \\
 ecc1 \\
 ecc0
 \end{pmatrix}
 \Rightarrow
 \begin{pmatrix}
 ecc8z \\
 ecc7 \\
 ecc6z \\
 ecc5 \\
 ecc4 \\
 ecc3z \\
 ecc2 \\
 ecc1z \\
 ecc0z
 \end{pmatrix}$$

Figure 31.3 Matrix for ECC Encoding

(2) Error Detection

According to the following formula, 9-bit syndromes (synd8 to synd0) are generated through 128-bit data (d127 to d0) and 9-bit error detecting code (ecc8z, ecc7, ecc6z, ecc5, ecc4, ecc3z, ecc2, ecc1z and ecc0z) which are read from the Code Flash.

$$\begin{pmatrix}
 10000010 & 10000100 & 10001111 & 11111111 & 11111111 & 11000000 & 00000000 & 00000000 & 00000000 & 00000111 & 10111011 & 01001000 & 01000100 & 10110010 & 00111110 & 01011101 \\
 01000001 & 01000010 & 10000111 & 11100000 & 00000000 & 00111111 & 11111111 & 10000000 & 00000000 & 00000111 & 01110110 & 10101000 & 10001001 & 01010011 & 00011111 & 00101111 \\
 10100000 & 00100001 & 01000100 & 00011111 & 00000000 & 00111110 & 00000000 & 01111111 & 11100000 & 00000110 & 11101101 & 10011001 & 00010010 & 01100011 & 10001111 & 10010111 \\
 01010000 & 10010000 & 01000010 & 00010000 & 11110000 & 00100001 & 11100000 & 01110000 & 00011111 & 10000101 & 11011100 & 01110010 & 00100011 & 10001111 & 11000111 & 11001010 \\
 00101000 & 01001000 & 00100001 & 00001000 & 10001110 & 00010001 & 00011100 & 01000111 & 00011100 & 01110011 & 11000011 & 11110100 & 00111100 & 00001111 & 11100001 & 11100101 \\
 00010100 & 00100100 & 00100000 & 10000100 & 01001001 & 10001000 & 10010011 & 00100100 & 11010011 & 01101000 & 00111111 & 11110111 & 11000000 & 00001101 & 11110010 & 11110010 \\
 00001010 & 00010010 & 00010000 & 01000010 & 00100101 & 01000100 & 01001010 & 10010010 & 10101010 & 11011000 & 00000000 & 00001111 & 11111111 & 11110100 & 11111001 & 01111001
 \end{pmatrix}
 \times
 \begin{pmatrix}
 d127 \\
 d126 \\
 : \\
 d1 \\
 d0 \\
 !ecc8z \\
 ecc7 \\
 !ecc6z \\
 ecc5 \\
 ecc4 \\
 !ecc3z \\
 ecc2 \\
 !ecc1z \\
 !ecc0z
 \end{pmatrix}
 =
 \begin{pmatrix}
 synd8 \\
 synd7 \\
 synd6 \\
 synd5 \\
 synd4 \\
 synd3 \\
 synd2 \\
 synd1 \\
 synd0
 \end{pmatrix}$$

Figure 31.4 Matrix for ECC Decoding

When synd8 to synd0 are all 0s, error correction is not performed. When some of synd8 to synd0 are 1, the target bit for correction is identified according to **Table 31.140** and the error bit is corrected. When synd8 to synd0 have values not indicated in **Table 31.140**, 2-bit error is detected.

Table 31.140 Correspondence Between Correction Target and Syndrome (1/2)

| synd[8:0] | Error Bit Position | synd[8:0] | Error Bit Position | synd[8:0] | Error Bit Position |
|-----------|--------------------|-----------|--------------------|-----------|--------------------|
| 10000000 | ECC data bit 8 | 100010100 | Flash data bit 90 | 110101001 | Flash data bit 44 |
| 010000000 | ECC data bit 7 | 100010010 | Flash data bit 89 | 101101001 | Flash data bit 43 |
| 001000000 | ECC data bit 6 | 100001100 | Flash data bit 88 | 011101001 | Flash data bit 42 |
| 000100000 | ECC data bit 5 | 100001010 | Flash data bit 87 | 110011001 | Flash data bit 41 |
| 000010000 | ECC data bit 4 | 100000110 | Flash data bit 86 | 101011001 | Flash data bit 40 |
| 000001000 | ECC data bit 3 | 011100000 | Flash data bit 85 | 011011001 | Flash data bit 39 |
| 000000100 | ECC data bit 2 | 011010000 | Flash data bit 84 | 100111001 | Flash data bit 38 |
| 000000010 | ECC data bit 1 | 011001000 | Flash data bit 83 | 010111001 | Flash data bit 37 |
| 000000001 | ECC data bit 0 | 011000100 | Flash data bit 82 | 001111001 | Flash data bit 36 |
| 101000001 | Flash data bit 127 | 011000010 | Flash data bit 81 | 111000101 | Flash data bit 35 |
| 010100001 | Flash data bit 126 | 010110000 | Flash data bit 80 | 000011111 | Flash data bit 34 |
| 001010001 | Flash data bit 125 | 010101000 | Flash data bit 79 | 000101111 | Flash data bit 33 |
| 000101001 | Flash data bit 124 | 010100100 | Flash data bit 78 | 001001111 | Flash data bit 32 |
| 000010101 | Flash data bit 123 | 010100010 | Flash data bit 77 | 010001111 | Flash data bit 31 |
| 000001011 | Flash data bit 122 | 010011000 | Flash data bit 76 | 100001111 | Flash data bit 30 |
| 100000101 | Flash data bit 121 | 010010100 | Flash data bit 75 | 000110111 | Flash data bit 29 |
| 010000011 | Flash data bit 120 | 010010010 | Flash data bit 74 | 001010111 | Flash data bit 28 |
| 100100001 | Flash data bit 119 | 010001100 | Flash data bit 73 | 010010111 | Flash data bit 27 |
| 010010001 | Flash data bit 118 | 010001010 | Flash data bit 72 | 100010111 | Flash data bit 26 |
| 001001001 | Flash data bit 117 | 010000110 | Flash data bit 71 | 001100111 | Flash data bit 25 |
| 000100101 | Flash data bit 116 | 001110000 | Flash data bit 70 | 010100111 | Flash data bit 24 |
| 000010011 | Flash data bit 115 | 001101000 | Flash data bit 69 | 100100111 | Flash data bit 23 |
| 100001001 | Flash data bit 114 | 001100100 | Flash data bit 68 | 011000111 | Flash data bit 22 |
| 010000101 | Flash data bit 113 | 001100010 | Flash data bit 67 | 101000111 | Flash data bit 21 |
| 001000011 | Flash data bit 112 | 001011000 | Flash data bit 66 | 110000111 | Flash data bit 20 |
| 110000001 | Flash data bit 111 | 001010100 | Flash data bit 65 | 000111011 | Flash data bit 19 |
| 001100001 | Flash data bit 110 | 001010010 | Flash data bit 64 | 000111101 | Flash data bit 18 |
| 000011001 | Flash data bit 109 | 001001100 | Flash data bit 63 | 111110000 | Flash data bit 17 |
| 000000111 | Flash data bit 108 | 001001010 | Flash data bit 62 | 011111000 | Flash data bit 16 |
| 100000011 | Flash data bit 107 | 001000110 | Flash data bit 61 | 001111100 | Flash data bit 15 |
| 111000000 | Flash data bit 106 | 000111000 | Flash data bit 60 | 000111110 | Flash data bit 14 |
| 110100000 | Flash data bit 105 | 000110100 | Flash data bit 59 | 100011110 | Flash data bit 13 |
| 110010000 | Flash data bit 104 | 000110010 | Flash data bit 58 | 110001110 | Flash data bit 12 |
| 110001000 | Flash data bit 103 | 000101100 | Flash data bit 57 | 111000110 | Flash data bit 11 |
| 110000100 | Flash data bit 102 | 000101010 | Flash data bit 56 | 111100010 | Flash data bit 10 |
| 110000010 | Flash data bit 101 | 000100110 | Flash data bit 55 | 111101000 | Flash data bit 9 |
| 101100000 | Flash data bit 100 | 000011100 | Flash data bit 54 | 011110100 | Flash data bit 8 |
| 101010000 | Flash data bit 99 | 000011010 | Flash data bit 53 | 001111010 | Flash data bit 7 |
| 101001000 | Flash data bit 98 | 000010110 | Flash data bit 52 | 100111100 | Flash data bit 6 |
| 101000100 | Flash data bit 97 | 000001110 | Flash data bit 51 | 010011110 | Flash data bit 5 |
| 101000010 | Flash data bit 96 | 111100001 | Flash data bit 50 | 101001110 | Flash data bit 4 |

Table 31.140 Correspondence Between Correction Target and Syndrome (2/2)

| synd[8:0] | Error Bit Position | synd[8:0] | Error Bit Position | synd[8:0] | Error Bit Position |
|-----------|--------------------|-----------|--------------------|-----------|--------------------|
| 100110000 | Flash data bit 95 | 111010001 | Flash data bit 49 | 110100110 | Flash data bit 3 |
| 100101000 | Flash data bit 94 | 110110001 | Flash data bit 48 | 111010010 | Flash data bit 2 |
| 100100100 | Flash data bit 93 | 101110001 | Flash data bit 47 | 011101010 | Flash data bit 1 |
| 100100010 | Flash data bit 92 | 011110001 | Flash data bit 46 | 111010100 | Flash data bit 0 |
| 100011000 | Flash data bit 91 | 111001001 | Flash data bit 45 | 000000000 | No error |

31.2.10.4 ECC Function for Data Flash

(1) Generating an Error Correcting Code (ECC)

The following formulas are used to generate 7-bit error detecting codes (ecc6z to ecc0z) for 32-bit data (d31 to d0). All bits of error correction code are inverted. With this measure, all 0s of data and error correction code become an illegal combination.

- $\text{ecc6z} = \text{!(d13} \wedge \text{d12} \wedge \text{d11} \wedge \text{d10} \wedge \text{d9} \wedge \text{d8} \wedge \text{d7} \wedge \text{d6} \wedge \text{d5} \wedge \text{d4} \wedge \text{d3} \wedge \text{d2} \wedge \text{d1} \wedge \text{d0})$
- $\text{ecc5z} = \text{!(d23} \wedge \text{d22} \wedge \text{d21} \wedge \text{d20} \wedge \text{d19} \wedge \text{d18} \wedge \text{d17} \wedge \text{d16} \wedge \text{d15} \wedge \text{d14} \wedge \text{d3} \wedge \text{d2} \wedge \text{d1} \wedge \text{d0})$
- $\text{ecc4z} = \text{!(d29} \wedge \text{d28} \wedge \text{d27} \wedge \text{d26} \wedge \text{d25} \wedge \text{d24} \wedge \text{d17} \wedge \text{d16} \wedge \text{d15} \wedge \text{d14} \wedge \text{d7} \wedge \text{d6} \wedge \text{d5} \wedge \text{d4})$
- $\text{ecc3z} = \text{!(d31} \wedge \text{d30} \wedge \text{d26} \wedge \text{d25} \wedge \text{d24} \wedge \text{d20} \wedge \text{d19} \wedge \text{d18} \wedge \text{d14} \wedge \text{d10} \wedge \text{d9} \wedge \text{d8} \wedge \text{d4} \wedge \text{d0})$
- $\text{ecc2z} = \text{!(d31} \wedge \text{d30} \wedge \text{d28} \wedge \text{d27} \wedge \text{d24} \wedge \text{d22} \wedge \text{d21} \wedge \text{d18} \wedge \text{d15} \wedge \text{d12} \wedge \text{d11} \wedge \text{d8} \wedge \text{d5} \wedge \text{d1})$
- $\text{ecc1z} = \text{!(d30} \wedge \text{d29} \wedge \text{d27} \wedge \text{d25} \wedge \text{d23} \wedge \text{d21} \wedge \text{d19} \wedge \text{d16} \wedge \text{d13} \wedge \text{d11} \wedge \text{d9} \wedge \text{d6} \wedge \text{d2} \wedge \text{d0})$
- $\text{ecc0z} = \text{!(d31} \wedge \text{d29} \wedge \text{d28} \wedge \text{d26} \wedge \text{d23} \wedge \text{d22} \wedge \text{d20} \wedge \text{d17} \wedge \text{d13} \wedge \text{d12} \wedge \text{d10} \wedge \text{d7} \wedge \text{d3} \wedge \text{d0})$

Where \wedge denotes an Exclusive OR.

(2) Error Detection

According to the following formula, 7-bit syndromes (synd6 to synd0) are generated through 32-bit data (d31 to d0) and 7-bit error detecting code (ecc6z to ecc0z) which are read from the Data Flash.

$$\begin{pmatrix} 1000000 & 00000000 & 00000000 & 00111111 & 11111111 \\ 0100000 & 00000000 & 11111111 & 11000000 & 00001111 \\ 0010000 & 00111111 & 00000011 & 11000000 & 11110000 \\ 0001000 & 11000111 & 00011100 & 01000111 & 00010001 \\ 0000100 & 11011001 & 01100100 & 10011001 & 00100010 \\ 0000010 & 01101010 & 10101001 & 00101010 & 01000101 \\ 0000001 & 10110100 & 11010010 & 00110100 & 10001001 \end{pmatrix} \times \begin{pmatrix} \text{!ecc6z} \\ \text{!ecc5z} \\ : \\ \text{!ecc1z} \\ \text{!ecc0z} \\ \text{d31} \\ \text{d30} \\ : \\ \text{d1} \\ \text{d0} \end{pmatrix} = \begin{pmatrix} \text{synd6} \\ \text{synd5} \\ \text{synd4} \\ \text{synd3} \\ \text{synd2} \\ \text{synd1} \\ \text{synd0} \end{pmatrix}$$

Figure 31.5 Matrix for EDC Decoding

When synd6 to synd0 are all 0s, error correction is not performed. When some of synd6 to synd0 are 1, the target bit for correction is identified according to **Table 31.141** and the error bit is corrected. When synd6 to synd0 have values not indicated in **Table 31.141**, 2-bit error is detected.

Table 31.141 Correspondence between Correction Target and Syndrome (1/2)

| synd[6:0] | Error Bit Position | synd[6:0] | Error Bit Position |
|-----------|--------------------|-----------|--------------------|
| 1000000 | ECC data bit 6 | 0101100 | FLASH data bit 18 |
| 0100000 | ECC data bit 5 | 0110001 | FLASH data bit 17 |
| 0010000 | ECC data bit 4 | 0110010 | FLASH data bit 16 |
| 0001000 | ECC data bit 3 | 0110100 | FLASH data bit 15 |
| 0000100 | ECC data bit 2 | 0111000 | FLASH data bit 14 |

Table 31.141 Correspondence between Correction Target and Syndrome (2/2)

| synd[6:0] | Error Bit Position | synd[6:0] | Error Bit Position |
|------------------|---------------------------|------------------|---------------------------|
| 0000010 | ECC data bit 1 | 1000011 | FLASH data bit 13 |
| 0000001 | ECC data bit 0 | 1000101 | FLASH data bit 12 |
| 0001101 | FLASH data bit 31 | 1000110 | FLASH data bit 11 |
| 0001110 | FLASH data bit 30 | 1001001 | FLASH data bit 10 |
| 0010011 | FLASH data bit 29 | 1001010 | FLASH data bit 9 |
| 0010101 | FLASH data bit 28 | 1001100 | FLASH data bit 8 |
| 0010110 | FLASH data bit 27 | 1010001 | FLASH data bit 7 |
| 0011001 | FLASH data bit 26 | 1010010 | FLASH data bit 6 |
| 0011010 | FLASH data bit 25 | 1010100 | FLASH data bit 5 |
| 0011100 | FLASH data bit 24 | 1011000 | FLASH data bit 4 |
| 0100011 | FLASH data bit 23 | 1100001 | FLASH data bit 3 |
| 0100101 | FLASH data bit 22 | 1100010 | FLASH data bit 2 |
| 0100110 | FLASH data bit 21 | 1100100 | FLASH data bit 1 |
| 0101001 | FLASH data bit 20 | 1101011 | FLASH data bit 0 |
| 0101010 | FLASH data bit 19 | 0000000 | No error |

31.3 Lockstep

This product incorporates the lockstep function in CPU1 and DMA to quickly detect failures without software interaction. The CPU1 and DMA executes the program using two different circuits, that is, master core and checker core, and constantly compares the execution results of the two circuits. When the results do not match, an error in either of the circuits is assumed and an error notification is sent to the ECM.

The lockstep function of the CPU1 and DMA features failure insertion, with which errors can be intentionally caused and thus enabling self-diagnosis of the lockstep operation.

31.3.1 List of Registers

Table 31.142 List of Registers

| Address | Symbol | Register Name | R/W | Value after Reset | Access Size |
|------------------------|-----------------|--|-----|------------------------|-------------|
| FFFE ED00 _H | CMPTST0 | Comparator test register 0 | R/W | 0000 0000 _H | 8/16/32 |
| FFFE ED04 _H | CMPTST1 | Comparator test register 1 | R/W | 0000 0000 _H | 8/16/32 |
| FFC4 CA00 _H | PDMA_COMP_CNTRL | PDMA Comparator Error Injection Control Register | R/W | 0000 0000 _H | 32 |

CMPTST0 and CMPTST1 are placed in the CPU Peripheral of the CPU1. These registers can only be accessed by the CPU1.

31.3.2 Details of Registers

31.3.2.1 CMPTST0 — Comparator Test Register 0

CMPTST0 is test register 0 used for the lockstep function of the CPU1.

Combining CMPTST0 with CMPTST1 enables self-diagnosis of the lockstep function. The following gives an example of self-diagnosis procedure.

- (1) Write arbitrary value to CMPTST0.
- (2) Write a different value to CMPTST1.
- (3) Read CMPTST0. The different values are read and sent to the master core and checker core.
- (4) Using the values read, run the comparator to be diagnosed.

Access: CMPTST0 can be read/written in 32/16/8-bit units.

Address: FFFE ED00_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | CMPTST0[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CMPTST0[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.143 CMPTST0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31 to 0 | CMPTST0[31:0] | Write: Data is written to each byte. Read: PE1: CMPTST0[31:0] value is read. PE1C: CMPTST1[31:0] value is read. |

31.3.2.2 CMPTST1 — Comparator Test Register 1

CMPTST1 is test register 1 used for the lockstep function of the CPU1.

Combining CMPTST1 with CMPTST0 enables self-diagnosis of the lockstep function.

Access: CMPTST1 can be read/written in 32/16/8-bit units.

Address: FFFE ED04_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | CMPTST1[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CMPTST1[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.144 CMPTST1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|---|
| 31 to 0 | CMPTST1[31:0] | Write: Data is written to each byte. Read: PE1: CMPTST1[31:0] value is read. PE1C: CMPTST0[31:0] value is read. |

31.3.2.3 PDMA_COMP_CNTRL — PDMA Comparator Error Injection Control Register

This register (PDMA_COMP_CNTRL) can control the output signals on the checker side of the DMA.

A DMA comparison error can be generated by configuring this register.

Access: PDMA_COMP_CNTRL can be read/written in 32-bit units.

Address: FFC4 CA00_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|-----|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | PROT1 PROT0 | | DMACMPERR[29:16] | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DMACMPERR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.145 PDMA_COMP_CNTRL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 31, 30 | PROT[1:0] | Protection bit 10: write is enable Other: write is disable These bits are always read as 0. |
| 29 to 0 | DMACMPERR [29:0] | A DMA comparison error can be generated by writing 111111_11111111_01111111_01111111 to DMACMPERR simultaneously with the PROT bit. Clear all these bits to 0 if there is no need to generate this error. |

31.3.3 Usage Notes

Reading a register whose value is undefined after reset without initializing it may lead to a CPU comparison error. Accordingly, we recommend initializing such registers with the desired settings.

CAUTION

Be sure to use the startup routine to initialize these registers before referring to them since they can be implicitly used by the C compiler.

Table 31.146 List of Registers that Need Initialization to Prevent Comparison Error

| Type | Register |
|------------------------------------|---|
| General purpose registers | r1 to r31 |
| Basic System registers | EIPC, FEPC, CTPC, CTBP, EIWR, FEWR, EBASE, INTBP, SCCFG, SCBP, MEA, ASID, MEI |
| FPU System Registers* ¹ | FPSR, FPEPC, FPST, FPCC |
| MPU Function System Registers | MCA, MCS, MCR, MPLA0 to MPLA15, MPUA0 to MPUA15, MPAT0 to MPAT15 |
| Cache Operation Function Registers | ICTAGL, ICTAGH, ICDATL, ICDATH, ICERR |

Note 1. When FPU is enabled, undefined bits of these register need to be initialized to 0.

31.4 Memory Protection

31.4.1 Overview

The overall memory protection architecture is shown in **Figure 31.6**. CPU1 has a Memory Protection Unit (MPU) that defines the access protection for the software. In addition, each resource (bus slave) has a guard mechanism that controls the access by any bus master, including ones that do not have a MPU such as the DMA or the FlexRay.

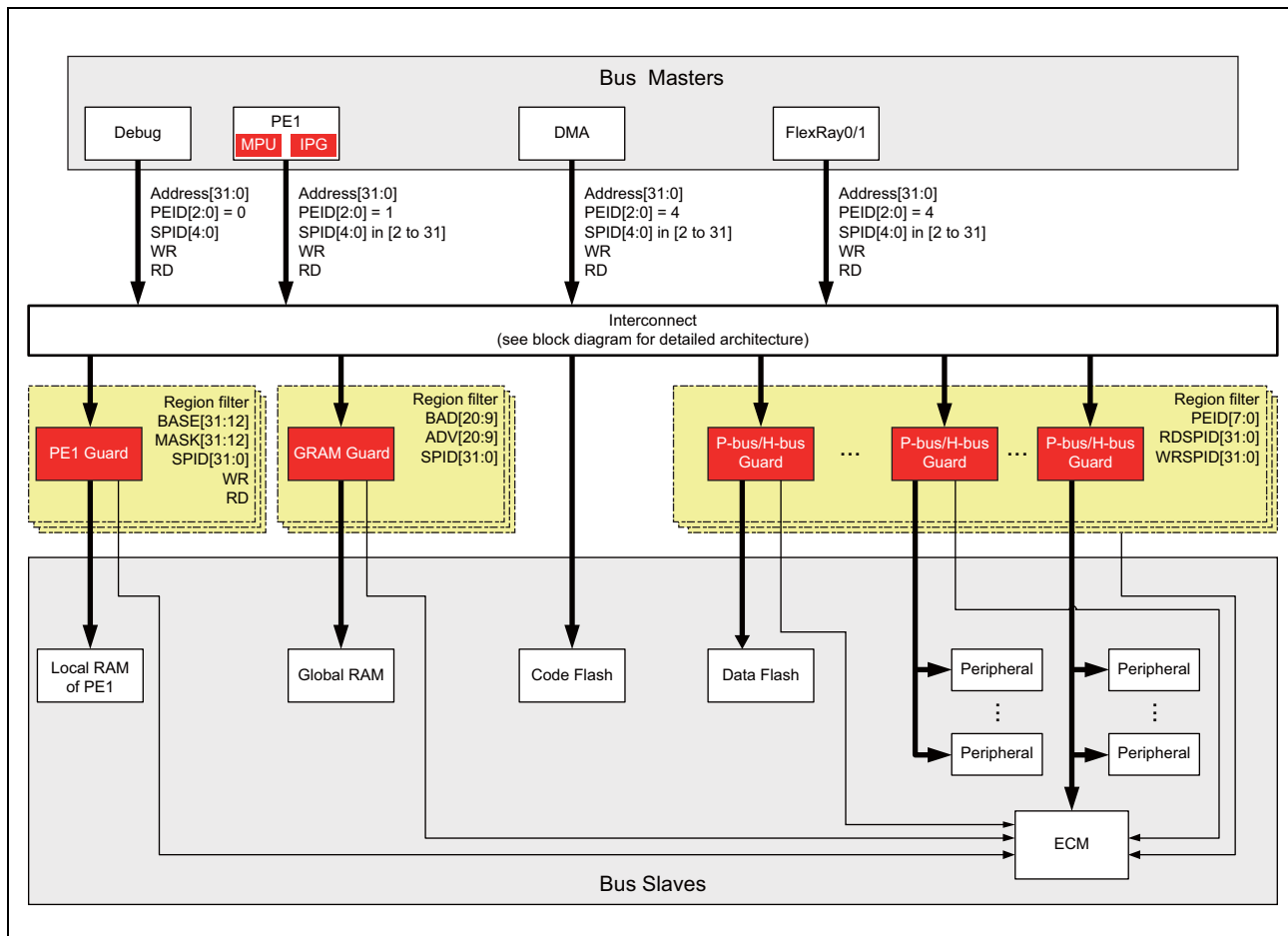


Figure 31.6 Architecture of the memory protection

For this purpose, each bus access contains the following context information that identifies the bus master.

- Address: Memory location to access
- PEID: Processor Element ID (PEID)
- SPID: System Protection ID of the bus master
- WR: Access is a write request if set to 1
- RD: Access is a read request if set to 1

Table 31.147 Bus Master Identifier

| Identifier | Function |
|------------|--|
| SPID | <p>When the CPU makes an access, the system protection identifier SPID that is assigned to the CPU is indicated.</p> <p>When the DMAC or DTS makes an access, the value of this identifier is the value in the channel master setting register.</p> <p>When other masters make an access, the value of this identifier is defined by each SPID register.</p> <p>As 0 and 1 are reserved, only 2 to 31 can be used.</p> |
| PEID | <p>The access source bus master is indicated.</p> <p>000_B: Reserved</p> <p>001_B: PE1</p> <p>010_B: Reserved</p> <p>011_B: Reserved</p> <p>100_B: DMAC,DTS, FlexRay</p> <p>101_B: Reserved</p> <p>110_B: Reserved</p> <p>111_B: Reserved</p> |

All cores can only use a value between 2 and 31.

For the peripherals that can act as bus masters (i.e., FlexRay), there are registers that can be used to configure and lock the SPID until next release. This register does not indicate the changed SPID value if the bus system overrides it due to a violation of the allowed range. This register can be read by the PE1, to check if the chosen SPID is correct.

The debug sub system can use any SPID. However, it cannot interfere with any function during normal operation, as it requires a secure handshake for activation.

Finally, the SPID that the DMA core uses can also be configured for each DMA/DTS channel by changing the channel master setting registers DMxCM/DTSmCM. For all DMxCM/DTSmCM registers, there exists one dedicated guard region. It can be used to prohibit write access to all channel master setting registers by any bus master after the initial configuration has been performed. Similar to the peripheral bus masters, the chosen SPID can be controlled if the guard region allows read access by the given bus master.

Guards control the access to the resources based on the specified bus access context information. Each guard supports a number of regions. Each region defines a filter specifying the bus context that can be accessed.

There exist different types of guards depending on the memory resource: local RAM, global RAM, peripheral registers and flash. Each of those memory resources is protected by a guard. Hence, no bus master can access a resource without proper configuration of the guard. All guards except the flash guards support access protection based on SPID. Hence, read and write accesses to peripherals whose registers are located in P-bus area and the H-bus area can be protected. All registers located in the Local Peripheral Bus (LPB) area cannot be accessed by an external bus master.

The guard setting registers are connected to the P-bus. Access to these registers is also protected by a P-bus guard. In order to prohibit reconfiguration of any guard in user mode, a dedicated SPID is reserved that is only used by the cores in supervisor mode. The SPIDs of the peripherals and the DMA can be locked and checked by hardware so that they do not use the supervisor SPID. Only PE1 can change the SPID of supervisor mode.

If a guard detects an illegal access, it reports it to the ECM. Additionally, information about the access context is stored in dedicated registers. The bus cycle is completed with an error response.

In case of an illegal write access, the data is not written. In case of an illegal read access, undefined data is returned.

31.4.1.1 Protection Mechanisms Inside the Processors

The PE has three mechanisms to protect its resources. The first mechanism is the Memory Protection Unit (MPU). It protects memory regions against illegal accesses by software. The second mechanism is the Internal Peripheral Guard (IPG). It protects registers of internal peripherals against accesses by software or external bus masters. The last mechanism is the PE Guard. It protects the whole PE against accesses by external bus masters.

(1) Memory Protection Unit (MPU)

Each CPU core has a dedicated Memory Protection Unit (MPU) to prevent unauthorized accesses to instruction and data. It is shown in red boxes with the label “MPU” in **Figure 31.6**. The MPU separates memory spaces, so that operation errors are constrained to the configured address space. Any unpermitted access to a protected memory area will raise a memory protection exception. As the MPU is integrated into the CPU core, it is aware of its pipeline and so it can generate precise exceptions on access violation.

If memory protection is enabled, all accesses which are not specifically enabled by an MPU area are prevented. The number of MPU areas is device dependent and generally between four and sixteen. Each MPU area is configured by writing to dedicated system registers. Configuration of the MPU can only be performed in supervisor mode. Each CPU can only access and configure its own MPU.

Reconfiguration of MPU areas is typically done by the scheduler during context switch. One area may be setup to watch for stack overflows and underflows. Some MPU settings usually apply globally to all threads and therefore these settings are not changed by the scheduler.

The CPU can operate in two privilege modes, user mode and supervisor mode. Privileges in user mode are restricted, while the supervisor mode permits all operations and accesses. There may be restrictions even in supervisor mode, but the CPU has the privilege to remove them if required. User mode does not permit to remove any restrictions.

The MPU provides up to 16 individual unified protection areas. The start and end address of each area can be specified with a granularity of one word (32-bit) and the size of each area may cover the whole address space. The access permissions for each area can be set to permit or prevent read accesses, write accesses and/or execute accesses. These access privileges can be specified independently for user mode and for supervisor mode. Areas may overlap, in which case permissions take precedence, i.e. if one of the areas permits the access, it is granted.

If the MPU detects an access which is not permitted, the access is not executed and a precise exception is raised. The exception handler can then decide to stop the violating thread or remove the restriction and resume execution.

The MPU only checks the addresses of the instruction and involved data for the currently executed instruction. Therefore, speculative access to instructions or data (e.g., during a cache prefetch) can only trigger an illegal access if the instruction or data is used during instruction execution. All addresses that are not used by the CPU pipeline are also not checked by the MPU.

The peripheral address space is also part of the address space which is monitored by the MPU. Therefore the MPU must protect or not protect a specific peripheral address space as needed. However, the MPU monitors only memory accesses of its own CPU core, not the accesses of other bus masters.

For a detailed functional description of the MPU, refer to the description of PE1 in **Section 3, CPU System**.

(2) Internal Peripheral Guard (IPG)

PE1 has an internal peripheral guard (IPG) that protects the registers of peripherals inside the PE against invalid accesses. It is shown in red boxes with the label “IPG” in **Figure 31.6**. The protection granularity is based on a fixed number of peripheral IP groups. The assignment of peripherals to a group is also fixed.

The configuration is done via six registers IPGENUM and IPGPMTUM0 to 4.

If the guard detects an illegal access, it does not forward it to the peripheral. But the access context is stored in the two registers IPGECRUM and IPGADRUM. Additionally, a SYSERR exception is generated with the exception source code 0x18. The IPGECRUM and IPGADRUM registers can be reset by writing a 0 to them.

For a detailed functional description of the IPG, refer to the description of PE1 in **Section 3, CPU System**.

(3) PE Guard

Each PE has a guard that controls the access to the local RAM by other bus masters. It is shown as a red box with the label “PE1 Guard” in **Figure 31.6**. The access by bus masters can be controlled via their bus context for eight regions. Bus masters can only access the local RAM area if permission is granted. The address range and allowed bus masters of the region n are defined by the three registers PEGGnBA, PEGGnSP and PEGGnMK. The PEGGnBA register defines the base address of the region n. The PEGGnMK register defines which bits of PEGGnBA are compared with the access address. If MASKm bit is "0", BASEm bit is compared with bit m of the access address. Otherwise, the bit is ignored during the access check. Expressed in C language notation, the access address lies within the region if $(\text{access address} \& (\sim \text{MASK}) \& 0\text{xFFFFFF00H}) == (\text{BASE} \& (\sim \text{MASK}) \& 0\text{xFFFFFF00H})$. Please note that the lower bit 11 to 0 of BASE and MASK are always 0.

The PEGGnSP register defines the SPIDs that are allowed to access. Each set bit k of PEGGnSP register allows the access to the region n by the SPID k. Setting more than one bit allows to enable more than one SPID value at a time. For example, setting SPID to 5_H allows access with SPID = 0 and SPID = 2. The bits WR and RD of PEGGnBA control if write access or read access are allowed for region n. Finally, the bit EN decides if the region is considered during the access.

In case regions have overlapping address ranges, the resulting access permission is a union of the individual permissions. For example, if one region allows access for SPID 1 only, and the other one for SPID 2 only, the overlapping address ranges can be accessed by SPID 1 and 2.

The LOCK bit of PEGGnBA controls each region n individually if the registers PEGGnBA, PEGGnMK and PEGGnSP of region n can be changed. After reset, the registers can be re-written as long as LOCK remains zero. Once a 1 is written, all further write accesses are ignored. Reconfiguration is only possible after the next reset. Access protection of the PE-Guard registers can be done via the LOCK bit, the MPU and the IPG.

If the guard detects an illegal access, it reports the violation to the Error Control Module (ECM) and reports the details about the access in three registers ERRSTATCTL, ERRSTAT, and ERRINFO. No exception is occurred. The register ERRSTAT indicates if an access violation has occurred. The register ERRINFO describes the context of the first access that triggered a violation. Finally, the register ERRSTATCTL is used to reset the ERRSTAT register. Please note that any further access violations are not stored as long as the ERR bit of ERRSTAT is set to 1.

For a detailed functional description of the PE Guard, refer to the description of PE1 in **Section 3, CPU System**.

31.4.1.2 Peripheral Register Protection with H-bus Guard (HBG) and P-bus Guard (PBG)

Each bus has slave guards that control which core can access the bus. The slave guards are shown in red boxes with the label H-bus Guard (HBG) and P-bus Guard (PBG) in **Figure 31.6**. Both types of slave guards use the same register interface. The region of protection is a peripheral instance, i.e., the configuration applies to all registers of the given peripheral instance. For each region n , there exists for each direction d ($d = \text{even number for reads or } d = \text{odd number for writes}$) registers FSGDxPROTnd that configure the protection. The LOCK bit controls if these registers can be changed. After reset, the registers can be re-written as long as LOCK remains 0. Once a 1 is written, all further write accesses are ignored. Reconfiguration is only possible after the next reset.

The guard uses two filters in sequence to check if a core is allowed to access. The first filter checks the PEID. Each set bit k of $\text{PEID}[k]$ allows access to the peripheral by the PEID k . After this filter is passed, the guard checks if the SPID matches the SPID filter. Each set bit k of $\text{SPID}[k]$ allows access by SPID k . It is possible to set more than one bit in all filters, to enable more than one SPID/PEID value at a time. For example, setting PEID to 6_H allows access with PEID = 1 and PEID = 2.

If a guard detects an illegal access, it reports the violation to the Error Control Module (ECM) and reports the details about the access in three registers ERRSLVxCTL , ERRSLVxSTAT , and ERRSLVxTYPE . Depending on the bus topology, there exist multiple sets x of these three registers. The register ERRSLVxSTAT indicates if an access violation has occurred. The register ERRSLVxTYPE describes the context of the first access that triggered a violation. Finally, the register ERRSLVxCTL is used to reset the ERRSLVxSTAT register. Please note that any further access violations are not stored in ERRSLVxTYPE as long as the ERR bit of ERRSLVxSTAT is set to 1.

For a detailed functional description of the peripheral register protection, please refer to **31.4.3, PBG** and **31.4.4, HBG**.

31.4.1.3 Global RAM Protection

The global RAM is protected by a memory guard. It is shown as a red box with the label “GRAM Guard” in **Figure 31.6**. The guard can only protect against write accesses. Read accesses are always allowed by any bus master. The guard has eight memory regions that can be protected. The granularity of each region is 512 bytes. Each device has sufficient RAM to allow a fine granular control of the region size. For each region n , there exists four configuration registers MGDGRPROTn , MGDGRSPIDn , MGDGRBADn and MGDGRADVn . The register MGDGRPROTn configures the contexts that are allowed to access. The LOCK bit controls if the registers of region n can be changed. After reset, the four registers can be re-written as long as LOCK remains 0. Once a 1 is written, all further write accesses are ignored. Reconfiguration is only possible after the next reset. The bit EN activates the protection for the given region n .

Each set bit k of MGDGRSPIDn allows the access to the peripheral by the SPID k . Setting more than one bit allows to enable more than one SPID value at a time. For example, setting SPID to 5_H allows access with SPID = 0 and SPID = 2. The UM bit defines for which modes the access is allowed.

The address range of the region n is defined by the two registers MGDGRBADn and MGDGRADVn . The MGDGRBADn register defines the base address of the region n . The MGDGRADVn register defines which bits of MGDGRBADn are compared with the access address. If bit $\text{ADV}[m]$ is set, bit $\text{AD}[m]$ is compared with bit m of the access address. Otherwise, the bit is ignored during the access check. Expressed in C notation, the address lies within the region if $(\text{address}\&\text{ADV}) == (\text{AD}\&\text{ADV})$.

Access to memory region n is granted if the access context passes the filter defined by these four registers. In case regions have overlapping address ranges, each region will be individually guarded. For example, if one region allows access for SPID 1 only, and the other one for SPID 2 only, the overlapping address regions cannot be accessed neither by SPID 1 nor 2.

After reset, no region is enabled. As long as no region is enabled, the GRAM guard allows read/write access by any bus master with any SPID.

If the guard detects an illegal access, it forwards it to the Error Control Module (ECM). The access context is stored in three registers MGDGRSCTL, MGDGRSSTAT, and MGDGRSTYPE. The register MGDGRSSTAT indicates if an access violation has occurred. The register MGDGRSTYPE describes the context of the first access that triggered a violation. Finally, the register MGDGRSCTL is used to reset the MGDGRSSTAT register.

The registers of the memory guard are protected by a P-bus guard.

For a detailed functional description of the GRAM Guard, please refer to **31.4.2, GRG**.

31.4.1.4 Protectable Memory Regions

The number of supported memory regions and their granularity for each resource are summarized in **Table 31.148**.

Table 31.148 Protection Region and Granularity

| Protected resource | Protection mechanism | Number of memory regions | Minimum granularity of memory region [Byte] |
|---|----------------------|-------------------------------------|---|
| Software access to address space of PE1 | MPU | 16 | 4 |
| Access by other bus masters to local resources of PE1 | PE Guard | 8 | 4096 |
| Peripherals IP on P-bus and H-bus | P-bus / H-bus guard | See Section 4, Address Space | Peripheral instance (chip select) |
| Global RAM | GRAM Guard | 8 | 512 |

31.4.1.5 Default MPU and Guard Configuration

The MPU of PE is disabled after reset. The H-bus and P-bus guard allow read and write access by PE1 only. Any SPID value is allowed to access. All other guards require a configuration of the region first to define the protected address range. Therefore, their regions are disabled by default. See the reset values of the registers in the related sections for more information.

Please note that all guards can be reconfigured during runtime, as long as configuration is not prevented by a lock bit or a slave guard. The configuration of the PE guard, the GRAM guard and all P-bus guards can be changed while accesses are performed on the resource. The new configuration will not result in an undefined filter state. But the new filter configuration can become active at any time during an access to the protected resource. When there is no access to a protection target resource by guard, the guard setting has to change.

31.4.2 GRG

This product is provided with GRG, which is described in detail in the following sections.

31.4.2.1 List of Registers

Table 31.149 List of Registers (1/2)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------|------------|--|-----|------------------------|-------------|--------------|
| FFC49000H | MGDGRPROT0 | Global-RAM FS Guard Protection Setting Register0 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |
| FFC49004H | MGDGRSPID0 | Global-RAM FS Guard SPID Setting Register0 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49008H | MGDGRBAD0 | Global-RAM FS Guard Base Address Register0 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4900CH | MGDGRADV0 | Global-RAM FS Guard Address Valid Register0 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49010H | MGDGRPROT1 | Global-RAM FS Guard Protection Setting Register1 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |
| FFC49014H | MGDGRSPID1 | Global-RAM FS Guard SPID Setting Register1 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49018H | MGDGRBAD1 | Global-RAM FS Guard Base Address Register1 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4901CH | MGDGRADV1 | Global-RAM FS Guard Address Valid Register1 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49020H | MGDGRPROT2 | Global-RAM FS Guard Protection Setting Register2 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |
| FFC49024H | MGDGRSPID2 | Global-RAM FS Guard SPID Setting Register2 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49028H | MGDGRBAD2 | Global-RAM FS Guard Base Address Register2 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4902CH | MGDGRADV2 | Global-RAM FS Guard Address Valid Register2 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49030H | MGDGRPROT3 | Global-RAM FS Guard Protection Setting Register3 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |
| FFC49034H | MGDGRSPID3 | Global-RAM FS Guard SPID Setting Register3 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49038H | MGDGRBAD3 | Global-RAM FS Guard Base Address Register3 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4903CH | MGDGRADV3 | Global-RAM FS Guard Address Valid Register3 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49040H | MGDGRPROT4 | Global-RAM FS Guard Protection Setting Register4 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |
| FFC49044H | MGDGRSPID4 | Global-RAM FS Guard SPID Setting Register4 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49048H | MGDGRBAD4 | Global-RAM FS Guard Base Address Register4 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4904CH | MGDGRADV4 | Global-RAM FS Guard Address Valid Register4 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49050H | MGDGRPROT5 | Global-RAM FS Guard Protection Setting Register5 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |
| FFC49054H | MGDGRSPID5 | Global-RAM FS Guard SPID Setting Register5 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49058H | MGDGRBAD5 | Global-RAM FS Guard Base Address Register5 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4905CH | MGDGRADV5 | Global-RAM FS Guard Address Valid Register5 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49060H | MGDGRPROT6 | Global-RAM FS Guard Protection Setting Register6 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |

Table 31.149 List of Registers (2/2)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------|---------------------|---|-----|------------------------|-------------|--------------|
| FFC49064H | MGDGRSPID6 | Global-RAM FS Guard SPID Setting Register6 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49068H | MGDGRBAD6 | Global-RAM FS Guard Base Address Register6 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4906CH | MGDGRADV6 | Global-RAM FS Guard Address Valid Register6 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49070H | MGDGRPROT7 | Global-RAM FS Guard Protection Setting Register7 | R/W | 07FF FE10 _H | 8/16/32 | 31.4.2.2 (1) |
| FFC49074H | MGDGRSPID7 | Global-RAM FS Guard SPID Setting Register7 | R/W | FFFF FFFF _H | 32 | 31.4.2.2 (2) |
| FFC49078H | MGDGRBAD7 | Global-RAM FS Guard Base Address Register7 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (3) |
| FFC4907CH | MGDGRADV7 | Global-RAM FS Guard Address Valid Register7 | R/W | 0000 0000 _H | 8/16/32 | 31.4.2.2 (4) |
| FFC49100H | MGDGRSCTL_VCI2GRAM | Global-RAM FS Guard Control Register (VCI2GRAM) | R/W | 0000 0000 _H | 32 | 31.4.2.2 (5) |
| FFC49104H | MGDGRSSTAT_VCI2GRAM | Global-RAM FS Guard Error Status Register (VCI2GRAM) | R | 0000 0000 _H | 32 | 31.4.2.2 (6) |
| FFC4910CH | MGDGRSTYPE_VCI2GRAM | Global-RAM FS Guard Error Access Type Register (VCI2GRAM) | R | xxxx xxxx _H | 8/16/32 | 31.4.2.2 (7) |
| FFC49200H | MGDGRSCTL_PE1 | Global-RAM FS Guard Control Register (PE1) | R/W | 0000 0000 _H | 32 | 31.4.2.2 (5) |
| FFC49204H | MGDGRSSTAT_PE1 | Global-RAM FS Guard Error Status Register (PE1) | R | 0000 0000 _H | 32 | 31.4.2.2 (6) |
| FFC4920CH | MGDGRSTYPE_PE1 | Global-RAM FS Guard Error Access Type Register (PE1) | R | xxxx xxxx _H | 8/16/32 | 31.4.2.2 (7) |
| FFC49700H | MGDGRSCTL_AXI2GRAM | Global-RAM FS Guard Control Register (AXI2GRAM) | R/W | 0000 0000 _H | 32 | 31.4.2.2 (5) |
| FFC49704H | MGDGRSSTAT_AXI2GRAM | Global-RAM FS Guard Error Status Register (AXI2GRAM) | R | 0000 0000 _H | 32 | 31.4.2.2 (6) |
| FFC4970CH | MGDGRSTYPE_AXI2GRAM | Global-RAM FS Guard Error Access Type Register (AXI2GRAM) | R | xxxx xxxx _H | 8/16/32 | 31.4.2.2 (7) |

- MGDGRPROT_n, MGDGRSPID_n, MGDGRBAD_n, and MGDGRADV_n set the protection specifications for each channel (n: 0 to 7).
- MGDGRSCTL_*, MGDGRSSTAT_*, and MGDGRSTYPE_* indicate error information on each access port: “_VCI2GRAM” represents access from the system interconnect to the Global RAM, “_PE1” represents access from the CPU1 to the Global RAM, “_AXI2GRAM” represents access from the H-bus to the Global RAM,

31.4.2.2 Details of Registers

(1) MGDGRPROTn — Global-RAM FS Guard Protection Setting Register n (n = 0 to 7)

Access: These registers can be read/written in 32/16/8-bit units.

Address: MGDGRPROT0: FFC4 9000_H
 MGDGRPROT1: FFC4 9010_H
 MGDGRPROT2: FFC4 9020_H
 MGDGRPROT3: FFC4 9030_H
 MGDGRPROT4: FFC4 9040_H
 MGDGRPROT5: FFC4 9050_H
 MGDGRPROT6: FFC4 9060_H
 MGDGRPROT7: FFC4 9070_H

Value after reset: 07FF FE10_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | LOCK | EN | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.150 MGDGRPROTn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 | LOCK | Lock bit 0: The register can be written 1: Any further writes to MGDGRPROTn and MGDGRSPIDn are ignored. This bit can only be cleared by reset. |
| 30 | EN | Guard Enable 0: Guard is disabled 1: Guard is enabled |
| 29 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(2) MGDGRSPIDn — Global-RAM FS Guard SPID Setting Register n (n = 0 to 7)

Access: These registers can be read/written in 32-bit units.

Address: MGDGRSPID0: FFC4 9004_H
 MGDGRSPID1: FFC4 9014_H
 MGDGRSPID2: FFC4 9024_H
 MGDGRSPID3: FFC4 9034_H
 MGDGRSPID4: FFC4 9044_H
 MGDGRSPID5: FFC4 9054_H
 MGDGRSPID6: FFC4 9064_H
 MGDGRSPID7: FFC4 9074_H

Value after reset: FFFF FFFF_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | SPID[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SPID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.151 MGDGRSPIDn Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 0 | SPID[31:0] | In the case of SPID[x]=0: Access with SPID = x is not allowed. In the case of SPID[x]=1: Access with SPID = x is allowed. |

(3) MGDGRBADn — Global-RAM FS Guard Base Address Register n (n = 0 to 7)

Access: These registers can be read/written in 32/16/8-bit units.

Address: MGDGRBAD0: FFC4 9008_H
 MGDGRBAD1: FFC4 9018_H
 MGDGRBAD2: FFC4 9028_H
 MGDGRBAD3: FFC4 9038_H
 MGDGRBAD4: FFC4 9048_H
 MGDGRBAD5: FFC4 9058_H
 MGDGRBAD6: FFC4 9068_H
 MGDGRBAD7: FFC4 9078_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|-----|-----|-----|-----|-----|-----|----|----|----|----|-----------|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | AD[20:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | AD[15:9] | | | | | | | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R | R |

Table 31.152 MGDGRBADn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 9 | AD[20:9] | Compare Base Address |
| 8 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(4) MGDGRADVn — Global-RAM FS Guard Address Valid Register n (n = 0 to 7)

Access: These registers can be read/written in 32/16/8-bit units.

Address: MGDGRADV0: FFC4 900C_H
 MGDGRADV1: FFC4 901C_H
 MGDGRADV2: FFC4 902C_H
 MGDGRADV3: FFC4 903C_H
 MGDGRADV4: FFC4 904C_H
 MGDGRADV5: FFC4 905C_H
 MGDGRADV6: FFC4 906C_H
 MGDGRADV7: FFC4 907C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----|-----|-----|-----|-----|-----|----|----|----|----|------------|-----|-----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | ADV[20:16] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ADV[15:9] | | | | | | | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R | R |

Table 31.153 MGDGRADVn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 21 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 20 to 9 | ADV[20:9] | Compare Address Valid MGDGRADVn[x] = 0 indicates that the bit is not to be compared. Therefore, if MGDGRADVn[x] is all 0s, any address will fall into the scope of access protection. |
| 8 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(5) MGDGRSCTL_VCI2GRAM/PE1 /AXI2GRAM — Global-RAM FS Guard Control Register

Access: These registers can be read/written in 32-bit units.

Address: MGDGRSCTL_VCI2GRAM: FFC4 9100_H
 MGDGRSCTL_PE1: FFC4 9200_H
 MGDGRSCTL_AXI2GRAM: FFC4 9700_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLRO | CLRE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 31.154 MGDGRSCTL_VCI2GRAM/PE1 /AXI2GRAM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | CLRO | Clear the OVF bit of MGDGRSSTAT_VCI2GRAM/PE1 /AXI2GRAM by writing "1" to this bit. Read value: 0: Clearing is completed 1: Clearing is in progress |
| 0 | CLRE | Clear the ERR bit of MGDGRSSTAT_VCI2GRAM/PE1 /AXI2GRAM by writing "1" to this bit. Read value: 0: Clearing is completed 1: Clearing is in progress |

| CLRO | CLRE | Function |
|------|------|---|
| 0 | 0 | Both bits are not cleared |
| 0 | 1 | Both bits are not cleared (this setting is ignored) |
| 1 | 0 | OVF bit is cleared |
| 1 | 1 | Both bits are cleared |

(6) MGDGRSSTAT_VCI2GRAM/PE1 /AXI2GRAM — Global-RAM FS Guard Error Status Register

Access: These registers can be read only in 32-bit units.

Address: MGDGRSSTAT_VCI2GRAM: FFC4 9104_H
 MGDGRSSTAT_PE1: FFC4 9204_H
 MGDGRSSTAT_AXI2GRAM: FFC4 9704_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVF | ERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.155 MGDGRSSTAT_VCI2GRAM/PE1 /AXI2GRAM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1 | OVF | Error Overflow Status Flag 0:No Overflow 1:Error Overflow |
| 0 | ERR | Error Status Flag 0:No Error 1:Error |

(7) MGDGRSTYPE_VCI2GRAM/PE1 /AXI2GRAM — Global-RAM FS Guard Error Access Type Register

Access: These registers can be read only in 32/16/8-bit units.

Address: MGDGRSTYPE_VCI2GRAM: FFC4 910C_H
 MGDGRSTYPE_PE1: FFC4 920C_H
 MGDGRSTYPE_AXI2GRAM: FFC4 970C_H

Value after reset: Undefined

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|----|----|----------|----|----|----|-----------|----|----|----|----|----|----|----|----|
| | — | — | — | ERRCAUSE | — | — | — | SPID[4:0] | | | | — | — | — | — | |
| Value after reset | — | — | — | 0 | — | — | — | 0 | 0 | 0 | 0 | 0 | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PEID[2:0] | | | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.156 MGDGRSTYPE_VCI2GRAM/PE1 /AXI2GRAM Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 29 | Reserved | These bits are undefined. |
| 28 | ERRCAUSE | This bit indicates if error cause was guard or illegal access 0: guard error 1: access to unmapped area |
| 27 to 25 | Reserved | These bits are undefined. |
| 24 to 20 | SPID[4:0] | SPID guard error / access to unmapped area error has occurred |
| 19 to 16 | Reserved | These bits are undefined. |
| 15 to 13 | PEID[2:0] | PEID guard error / access to unmapped area error has occurred |
| 12 to 0 | Reserved | These bits are undefined. |

31.4.3 PBG

The PBG module is divided into several PBG groups, each of which is provided with a maximum of 16 protection channels. A single PBG channel can designate the access against which a single peripheral circuit should be protected. Each PBG group can hold the information of the access that has been rejected.

The following table lists the peripheral circuits to be protected, the corresponding PBG group names, and the PBG channel numbers.

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (1/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|--------------|--------------------|------------------|--|--|---|
| APBGRD_PFSS1 | 0 | R | [P-bus guard] APBFSGDPROT_xxx_x APBFSGDSPID_xxx_x | APBFSGDPROT_PBG_A APBFSGDSPID_PBG_A | ERRSLVCTL_PFSS1 ERRSLVSTAT_PFSS1 ERRSLVTYPE_PFSS1 |
| | 1 | W | | APBFSGDPROT_PBG_B APBFSGDSPID_PBG_B | |
| | 2 | R | [Global RAM Guard] MGDGRSCTL_PE1 MGDGRSSTAT_PE1 MGDGRSTYPE_PE | APBFSGDPROT_SP1_A APBFSGDSPID_SP1_A | |
| | 3 | W | [Code Flash ECC] CFECCCTL_PE1 CFSTSTCTL_PE1 [Local RAM ECC] The entire module [Instruction Cache ECC and EDC] The entire module [ECC on Data Transfer] IFxxx_PE1 LSSxxx_PE1 LSMxxx_PE1 [PE Guard] PGERRSTATCTL_PE1 PGERRSTAT_PE1 PGERRINFO_PE1 | APBFSGDPROT_SP1_B APBFSGDSPID_SP1_B | |
| | 4 | | Reserved | | |
| | 5 | | Reserved | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (2/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group | | |
|--------------|--------------------|------------------|---|---|---|--|---|
| APBGRD_PFSS1 | 6 | R | [Global RAM Guard] Other than "MGDGRSCTL_PE1, MGDGRSSTAT_PE1, MGDGRSTYPE_PE1" | APBFSGDPROT_SP4_A APBFSGDSPID_SP4_A | ERRSLVCTL_PFSS1 ERRSLVSTAT_PFSS1 ERRSLVTYPE_PFSS1 | | |
| | 7 | W | [Code Flash ECC] Other than "CFECCCTL_PE1, CFSTSTCTL_PE1" [Global RAM ECC] The entire module [ECC on Data Transfer] CFxxx_VCI2CFBB VPxxx_SGn (n=0, 1, 2, 5) VCxxx_PDMA APxxx_PFSS [Flash] BFASLR [P-bus guard] ERRSLVCTL_PFSS0 ERRSLVCTL_PFSS1 ERRSLVSTAT_PFSS0 ERRSLVSTAT_PFSS1 ERRSLVTYPE_PFSS0 ERRSLVTYPE_PFSS1 [Lockstep] PDMA_COMP_CNTRL | APBFSGDPROT_SP4_B APBFSGDSPID_SP4_B | | | |
| | 8 | | Reserved | | | | |
| | 9 | | Reserved | | | | |
| | 10 | | Reserved | | | | |
| | 11 | | Reserved | | | | |
| | 12 | | Reserved | | | | |
| | 13 | | Reserved | | | | |
| | 14 | | Reserved | | | | |
| | 15 | | Reserved | | | | |
| | APBGRD_PFSS0 | 0 | R | [DMA/DTS] DMxCM (x=00 to 07, 10 to 17) DTSmCM (m=000 to 127) | | APBFSGDPROT_PDMACM_A APBFSGDSPID_PDMACM_A | ERRSLVCTL_PFSS0 ERRSLVSTAT_PFSS0 ERRSLVTYPE_PFSS0 |
| | | 1 | W | | | APBFSGDPROT_PDMACM_B APBFSGDSPID_PDMACM_B | |
| | | 2 | R | [DMA/DTS] Other than "DMxCM (x=00 to 07, 10 to 17), DTSmCM (m=000 to 127)" | | APBFSGDPROT_PDMACH_A APBFSGDSPID_PDMACH_A | |
| | | 3 | W | | | APBFSGDPROT_PDMACH_B APBFSGDSPID_PDMACH_B | |
| | | 4 | R | [INTC2] Other than "FEINTF, FEINTFC" | | APBFSGDPROT_INTC2_A APBFSGDSPID_INTC2_A | |
| 5 | | W | | APBFSGDPROT_INTC2_B APBFSGDSPID_INTC2_B | | | |
| 6 | | | Reserved | | | | |
| 7 | | | Reserved | | | | |
| 8 | | | Reserved | | | | |
| 9 | | | Reserved | | | | |
| 10 | | | Reserved | | | | |
| 11 | | | Reserved | | | | |
| 12 | | | Reserved | | | | |
| 13 | | | Reserved | | | | |
| 14 | | | Reserved | | | | |
| 15 | | Reserved | | | | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (3/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|----------------------------|--|
| PBG0 #0 | 0 | R | [P-bus guard] FSGD0PROTnn FSGD0SPIDnn | FSGD0PROT00 FSGD0SPID00 | ERRSLV0CTL ERRSLV0STAT ERRSLV0TYPE |
| | 1 | W | ERRSLV0CTL ERRSLV0STAT ERRSLV0TYPE | FSGD0PROT01 FSGD0SPID01 | |
| | 2 | | Reserved | | |
| | 3 | | Reserved | | |
| | 4 | R | [Flash] EEP(Data Flash)*1 | FSGD0PROT04 FSGD0SPID04 | |
| | 5 | W | | FSGD0PROT05 FSGD0SPID05 | |
| | 6 | R | [Flash] EEPRDCYCL | FSGD0PROT06 FSGD0SPID06 | |
| | 7 | W | | FSGD0PROT07 FSGD0SPID07 | |
| | 8 | | Reserved | | |
| | 9 | | Reserved | | |
| | 10 | R | [Data Flash ECC] The entire module | FSGD0PROT10 FSGD0SPID10 | |
| | 11 | W | | FSGD0PROT11 FSGD0SPID11 | |
| | 12 | | Reserved | | |
| | 13 | | Reserved | | |
| | 14 | | Reserved | | |
| 15 | | Reserved | | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (4/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|--|------------------------------|--|
| PBG1 #0 | 0 | R | [P-bus guard] FSGD1xPROTnn (x=A to E, nn=00 to 15) | FSGD1APROT00 FSGD1ASPID00 | ERRSLV1CTL ERRSLV1STAT ERRSLV1TYPE |
| | 1 | W | FSGD1xSPIDnn (x=A to E, nn=00 to 15) ERRSLV1CTL ERRSLV1STAT ERRSLV1TYPE | FSGD1APROT01 FSGD1ASPID01 | |
| | 2 | R | [ECC on Data Transfer] APEC0ECCCTL APEC0ERRINT | FSGD1APROT02 FSGD1ASPID02 | |
| | 3 | W | APEC0STCLR APEC0OVFSTR APEC01STERSTR APEC01STEADR0 | FSGD1APROT03 FSGD1ASPID03 | |
| | 4 | R | [TAUJ0] The entire module | FSGD1APROT04 FSGD1ASPID04 | |
| | 5 | W | | FSGD1APROT05 FSGD1ASPID05 | |
| | 6 | R | [TAUJ1] The entire module | FSGD1APROT06 FSGD1ASPID06 | |
| | 7 | W | | FSGD1APROT07 FSGD1ASPID07 | |
| | 8 | R | [TAUJ2] The entire module | FSGD1APROT08 FSGD1ASPID08 | |
| | 9 | W | | FSGD1APROT09 FSGD1ASPID09 | |
| | 10 | R | [TAUD0] The entire module | FSGD1APROT10 FSGD1ASPID10 | |
| | 11 | W | | FSGD1APROT11 FSGD1ASPID11 | |
| | 12 | R | [TAUD1] The entire module | FSGD1APROT12 FSGD1ASPID12 | |
| | 13 | W | | FSGD1APROT13 FSGD1ASPID13 | |
| | 14 | R | [TAUD2] The entire module | FSGD1APROT14 FSGD1ASPID14 | |
| 15 | W | | FSGD1APROT15 FSGD1ASPID15 | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (5/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|------------------------------|--|
| PBG1 #1 | 0 | R | [OSTM0] The entire module | FSGD1BPROT00 FSGD1BSPID00 | ERRSLV1CTL ERRSLV1STAT ERRSLV1TYPE |
| | 1 | W | | FSGD1BPROT01 FSGD1BSPID01 | |
| | 2 | R | [OSTM1] The entire module | FSGD1BPROT02 FSGD1BSPID02 | |
| | 3 | W | | FSGD1BPROT03 FSGD1BSPID03 | |
| | 4 | R | [OSTM] IC0CKSEL0 | FSGD1BPROT04 FSGD1BSPID04 | |
| | 5 | W | | FSGD1BPROT05 FSGD1BSPID05 | |
| | 6 | R | [OSTM] IC0CKSEL1 | FSGD1BPROT06 FSGD1BSPID06 | |
| | 7 | W | | FSGD1BPROT07 FSGD1BSPID07 | |
| | 8 | R | [TAPA0] The entire module | FSGD1BPROT08 FSGD1BSPID08 | |
| | 9 | W | | FSGD1BPROT09 FSGD1BSPID09 | |
| | 10 | R | [TAPA1] The entire module | FSGD1BPROT10 FSGD1BSPID10 | |
| | 11 | W | | FSGD1BPROT11 FSGD1BSPID11 | |
| | 12 | R | [TAPA2] The entire module | FSGD1BPROT12 FSGD1BSPID12 | |
| | 13 | W | | FSGD1BPROT13 FSGD1BSPID13 | |
| | 14 | R | [TAPA3] The entire module | FSGD1BPROT14 FSGD1BSPID14 | |
| 15 | W | | FSGD1BPROT15 FSGD1BSPID15 | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (6/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------------------|--|------------------------------|--|
| PBG1 #2 | 0 | R | [PIC1A] Other than "POMONSEL, PIMONSEL, SELBSSER" | FSGD1CPROT00 FSGD1CSPID00 | ERRSLV1CTL ERRSLV1STAT ERRSLV1TYPE |
| | 1 | W | | FSGD1CPROT01 FSGD1CSPID01 | |
| | 2 | R | [TSG30] The entire module | FSGD1CPROT02 FSGD1CSPID02 | |
| | 3 | W | | FSGD1CPROT03 FSGD1CSPID03 | |
| | 4 | R | [TSG31] The entire module | FSGD1CPROT04 FSGD1CSPID04 | |
| | 5 | W | | FSGD1CPROT05 FSGD1CSPID05 | |
| | 6 | R | [ENCA0] The entire module | FSGD1CPROT06 FSGD1CSPID06 | |
| | 7 | W | | FSGD1CPROT07 FSGD1CSPID07 | |
| | 8 | R | [ENCA1] The entire module | FSGD1CPROT08 FSGD1CSPID08 | |
| | 9 | W | | FSGD1CPROT09 FSGD1CSPID09 | |
| | 10 | R | [PSI50] The entire module | FSGD1CPROT10 FSGD1CSPID10 | |
| | 11 | W | | FSGD1CPROT11 FSGD1CSPID11 | |
| | 12 | R | [PSI51] The entire module | FSGD1CPROT12 FSGD1CSPID12 | |
| | 13 | W | | FSGD1CPROT13 FSGD1CSPID13 | |
| | 14 | R | [PSI5] PSI5TSSEL | FSGD1CPROT14 FSGD1CSPID14 | |
| 15 | W | FSGD1CPROT15 FSGD1CSPID15 | | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (7/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|------------------------------|--|
| PBG1 #3 | 0 | R | [PIC2B] Other than "ADSYNCTRG" | FSGD1DPROT00 FSGD1DSPID00 | ERRSLV1CTL ERRSLV1STAT ERRSLV1TYPE |
| | 1 | W | | FSGD1DPROT01 FSGD1DSPID01 | |
| | 2 | R | [TPBA0] The entire module | FSGD1DPROT02 FSGD1DSPID02 | |
| | 3 | W | | FSGD1DPROT03 FSGD1DSPID03 | |
| | 4 | R | [TPBA1] The entire module | FSGD1DPROT04 FSGD1DSPID04 | |
| | 5 | W | | FSGD1DPROT05 FSGD1DSPID05 | |
| | 6 | R | [SCI30] The entire module | FSGD1DPROT06 FSGD1DSPID06 | |
| | 7 | W | | FSGD1DPROT07 FSGD1DSPID07 | |
| | 8 | R | [SCI31] The entire module | FSGD1DPROT08 FSGD1DSPID08 | |
| | 9 | W | | FSGD1DPROT09 FSGD1DSPID09 | |
| | 10 | R | [SCI32] The entire module | FSGD1DPROT10 FSGD1DSPID10 | |
| | 11 | W | | FSGD1DPROT11 FSGD1DSPID11 | |
| | 12 | R | [RLIN30] The entire module | FSGD1DPROT12 FSGD1DSPID12 | |
| | 13 | W | | FSGD1DPROT13 FSGD1DSPID13 | |
| | 14 | R | [RLIN31] The entire module | FSGD1DPROT14 FSGD1DSPID14 | |
| 15 | W | | FSGD1DPROT15 FSGD1DSPID15 | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (8/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|------------------------------|--|
| PBG1 #4 | 0 | R | [RSENT0] The entire module | FSGD1EPROT00 FSGD1ESPID00 | ERRSLV1CTL ERRSLV1STAT ERRSLV1TYPE |
| | 1 | W | | FSGD1EPROT01 FSGD1ESPID01 | |
| | 2 | R | [RSENT1] The entire module | FSGD1EPROT02 FSGD1ESPID02 | |
| | 3 | W | | FSGD1EPROT03 FSGD1ESPID03 | |
| | 4 | R | [RSENT2] The entire module | FSGD1EPROT04 FSGD1ESPID04 | |
| | 5 | W | | FSGD1EPROT05 FSGD1ESPID05 | |
| | 6 | R | [RSENT3] The entire module | FSGD1EPROT06 FSGD1ESPID06 | |
| | 7 | W | | FSGD1EPROT07 FSGD1ESPID07 | |
| | 8 | R | [RSENT4] The entire module | FSGD1EPROT08 FSGD1ESPID08 | |
| | 9 | W | | FSGD1EPROT09 FSGD1ESPID09 | |
| | 10 | R | [RSENT5] The entire module | FSGD1EPROT10 FSGD1ESPID10 | |
| | 11 | W | | FSGD1EPROT11 FSGD1ESPID11 | |
| | 12 | R | [RSENT] RSENTTSEL | FSGD1EPROT12 FSGD1ESPID12 | |
| | 13 | W | | FSGD1EPROT13 FSGD1ESPID13 | |
| | 14 | R | [PIC1A] POMONSEL PIMONSEL SELBSSER | FSGD1EPROT14 FSGD1ESPID14 | |
| 15 | W | | FSGD1EPROT15 FSGD1ESPID15 | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (9/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|---|---|------------------------------|--|
| PBG2 #0 | 0 | R | [P-bus guard] FSGD2xPROTnn (x=A to C) FSGD2xSPIDnn (x=A to C) | FSGD2APROT00 FSGD2ASPID00 | ERRSLV2CTL ERRSLV2STAT ERRSLV2TYPE |
| | 1 | W | ERRSLV2CTL ERRSLV2STAT ERRSLV2TYPE | FSGD2APROT01 FSGD2ASPID01 | |
| | 2 | R | [ECC on Data Transfer] APEC1ECCCTL APEC1ERRINT APEC1STCLR | FSGD2APROT02 FSGD2ASPID02 | |
| | 3 | W | APEC1OVFSTR APEC11STERSTR APEC11STEADR0 [ECC for Peripheral RAM] ECCFLX0xxx ECCFLX0Tnxxx (n=0, 1) ECCCSIHnxxx (n=0 to 3) ECCCANnxxx (n=0, 1) [H-bus guard] HSSPIDRG0 | FSGD2APROT03 FSGD2ASPID03 | |
| | 4 | R | [OSTM3] The entire module | FSGD2APROT04 FSGD2ASPID04 | |
| | 5 | W | | FSGD2APROT05 FSGD2ASPID05 | |
| | 6 | R | [OSTM4] The entire module | FSGD2APROT06 FSGD2ASPID06 | |
| | 7 | W | | FSGD2APROT07 FSGD2ASPID07 | |
| | 8 | R | [OSTM5] The entire module | FSGD2APROT08 FSGD2ASPID08 | |
| | 9 | W | | FSGD2APROT09 FSGD2ASPID09 | |
| | 10 | R | [OSTM6] The entire module | FSGD2APROT10 FSGD2ASPID10 | |
| | 11 | W | | FSGD2APROT11 FSGD2ASPID11 | |
| | 12 | R | [OSTM7] The entire module | FSGD2APROT12 FSGD2ASPID12 | |
| | 13 | W | | FSGD2APROT13 FSGD2ASPID13 | |
| | 14 | R | [H-bus guard] FSGDF0PROT00 FSGDF0SPID00 FSGDF0PROT01 FSGDF0SPID01 | FSGD2APROT14 FSGD2ASPID14 | |
| 15 | W | ERRSLVF0CTL ERRSLVF0STAT ERRSLVF0TYPE | FSGD2APROT15 FSGD2ASPID15 | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (10/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|------------------------------|--|
| PBG2 #1 | 0 | R | [CSIG0] The entire module | FSGD2BPROT00 FSGD2BSPID00 | ERRSLV2CTL ERRSLV2STAT ERRSLV2TYPE |
| | 1 | W | | FSGD2BPROT01 FSGD2BSPID01 | |
| | 2 | R | [CSIH0] The entire module | FSGD2BPROT02 FSGD2BSPID02 | |
| | 3 | W | | FSGD2BPROT03 FSGD2BSPID03 | |
| | 4 | R | [CSIH1] The entire module | FSGD2BPROT04 FSGD2BSPID04 | |
| | 5 | W | | FSGD2BPROT05 FSGD2BSPID05 | |
| | 6 | R | [CSIH2] The entire module | FSGD2BPROT06 FSGD2BSPID06 | |
| | 7 | W | | FSGD2BPROT07 FSGD2BSPID07 | |
| | 8 | R | [CSIH3] The entire module | FSGD2BPROT08 FSGD2BSPID08 | |
| | 9 | W | | FSGD2BPROT09 FSGD2BSPID09 | |
| | 10 | R | [CSIH] SELCSIHDMA | FSGD2BPROT10 FSGD2BSPID10 | |
| | 11 | W | | FSGD2BPROT11 FSGD2BSPID11 | |
| | 12 | R | [RS-CANFD0] Other than "RSCAN0CANFMDR, RSCFD0CANFMDR" | FSGD2BPROT12 FSGD2BSPID12 | |
| | 13 | W | | FSGD2BPROT13 FSGD2BSPID13 | |
| | 14 | | Reserved | | |
| 15 | | Reserved | | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (11/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|------------------------------|--|
| PBG2 #2 | 0 | R | [RS-CANFD0] RSCAN0CANFMDMR RSCFD0CANFMDMR | FSGD2CPROT00 FSGD2CSPID00 | ERRSLV2CTL ERRSLV2STAT ERRSLV2TYPE |
| | 1 | W | | FSGD2CPROT01 FSGD2CSPID01 | |
| | 2 | R | [DCRAn (n=0 to 3)] The entire module | FSGD2CPROT02 FSGD2CSPID02 | |
| | 3 | W | | FSGD2CPROT03 FSGD2CSPID03 | |
| | 4 | R | [ECM] The entire module | FSGD2CPROT04 FSGD2CSPID04 | |
| | 5 | W | | FSGD2CPROT05 FSGD2CSPID05 | |
| | 6 | R | [FEINT factor register] FEINTF FEINTFC | FSGD2CPROT06 FSGD2CSPID06 | |
| | 7 | W | | FSGD2CPROT07 FSGD2CSPID07 | |
| | 8 | R | [WDTA] The entire module | FSGD2CPROT08 FSGD2CSPID08 | |
| | 9 | W | | FSGD2CPROT09 FSGD2CSPID09 | |
| | 10 | R | [DMA] PINTn (n=0 to 7) PINTCLRn (n=0 to 7) | FSGD2CPROT10 FSGD2CSPID10 | |
| | 11 | W | DTSTRGSEL0 DTSTRGSEL1 DMACTRGSEL0 DMACTRGSEL1 | FSGD2CPROT11 FSGD2CSPID11 | |
| | 12 | | Reserved | | |
| | 13 | | Reserved | | |
| | 14 | | Reserved | | |
| 15 | | Reserved | | | |
| PBG5 #0 | 0 | R | [P-bus guard] FSGD5xPROTnn (x=A to C) FSGD5xSPIDnn (x=A to C) | FSGD5APROT00 FSGD5ASPID00 | ERRSLV5CTL ERRSLV5STAT ERRSLV5TYPE |
| | 1 | W | ERRSLV5CTL ERRSLV5STAT ERRSLV5TYPE | FSGD5APROT01 FSGD5ASPID01 | |
| | 2 | R | [ECC on Data Transfer] APEC2ECCCTL APEC2ERRINT APEC2STCLR APEC2OVFSTR APEC21STERSTR APEC21STEADR0 | FSGD5APROT02 FSGD5ASPID02 | |
| | 3 | W | [DNF] DNFAnCTL (n=0 to 43) DNFAnEN (n=0 to 43) DNFAnENL (n=0 to 43) [BIST] LBISTREF1 LBISTREF2 MBISTREF LBISTSIG1 LBISTSIG2 MBISTSIG MBISTFTAGL MBISTFTAGH BSEQ0ST BSEQ0STB BISTST | FSGD5APROT03 FSGD5ASPID03 | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (12/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|------------------------------|--|
| PBG5 #0 | 4 | R | [Reset] RESF RESFC | FSGD5APROT04 FSGD5ASPID04 | ERRSLV5CTL ERRSLV5STAT ERRSLV5TYPE |
| | 5 | W | SWSRESA0 SWARES0 RESC STAC_DRTSRAM STAC_GRAM STAC_LM0 STAC_LM10 [BIST] BSEQ0CTL [Clock] CLKD2DIV CLKD2STAT CLKD3DIV CLKD3STAT CKSC2C CKSC2S CKSC3C CKSC3S CKSC8C CKSC8S [CVM] CVMF CVMDE CVMDMASK CVMDIAG CVMMON CVMFC CVMDEW [CLMA] CLMA _n CTL0 (n=0 to3) CLMA _n CMPL (n=0 to3) CLMA _n CMPH (n=0 to3) CLMA _n PCMD (n=0 to3) CLMA _n PS (n=0 to3) CLMATEST CLMATESTS [Mode] MODE [DNF] DNFCKS100C DNFCSC100STAT DNFCKS101C DNFCSC101STAT DNFCKS106C DNFCSC106STAT DNFCKS107C DNFCSC107STAT DNFCKS108C DNFCSC108STAT DNFCKS109C DNFCSC109STAT DNFCKS110C DNFCSC110STAT DNFCKS112C DNFCSC112STAT DNFCKS113C DNFCSC113STAT DNFCKS114C DNFCSC114STAT DNFCKS104C DNFCSC104STAT DNFCKS105C DNFCSC105STAT DNFCKS102C DNFCSC102STAT [Flash] FHVE3 FHVE15 | FSGD5APROT05 FSGD5ASPID05 | |
| | 6 | R | [JTAG port] The entire module | FSGD5APROT06 FSGD5ASPID06 | |
| | 7 | W | | FSGD5APROT07 FSGD5ASPID07 | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (13/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------------------|---|------------------------------|--|
| PBG5 #0 | 8 | R | [Port group n (n=0 to 5)] The entire module | FSGD5APROT08 FSGD5ASPID08 | ERRSLV5CTL ERRSLV5STAT ERRSLV5TYPE |
| | 9 | W | | FSGD5APROT09 FSGD5ASPID09 | |
| | 10 | R | [FCLA0] The entire module | FSGD5APROT10 FSGD5ASPID10 | |
| | 11 | W | | FSGD5APROT11 FSGD5ASPID11 | |
| | 12 | R | [FCLA1] The entire module | FSGD5APROT12 FSGD5ASPID12 | |
| | 13 | W | | FSGD5APROT13 FSGD5ASPID13 | |
| | 14 | R | [FCLA2] The entire module | FSGD5APROT14 FSGD5ASPID14 | |
| | 15 | W | | FSGD5APROT15 FSGD5ASPID15 | |
| PBG5 #1 | 0 | R | [FCLA3] The entire module | FSGD5BPROT00 FSGD5BSPID00 | ERRSLV5CTL ERRSLV5STAT ERRSLV5TYPE |
| | 1 | W | | FSGD5BPROT01 FSGD5BSPID01 | |
| | 2 | R | [FCLA4] The entire module | FSGD5BPROT02 FSGD5BSPID02 | |
| | 3 | W | | FSGD5BPROT03 FSGD5BSPID03 | |
| | 4 | | Reserved | | |
| | 5 | | Reserved | | |
| | 6 | R | [ADCG0] The entire module | FSGD5BPROT06 FSGD5BSPID06 | |
| | 7 | W | | FSGD5BPROT07 FSGD5BSPID07 | |
| | 8 | R | [ADCG1] The entire module | FSGD5BPROT08 FSGD5BSPID08 | |
| | 9 | W | | FSGD5BPROT09 FSGD5BSPID09 | |
| | 10 | R | [PIC2B] ADSYNCTR | FSGD5BPROT10 FSGD5BSPID10 | |
| | 11 | W | | FSGD5BPROT11 FSGD5BSPID11 | |
| | 12 | R | [TSN0] TSN0CR TSN0STAT TSN0DIAG [SINT] SINTRn (n=0 to 4) [Flash] (See the Flash Memory User's Manual: Hardware Interface) SELFIDn (n=0 to 3) SELFIDST | FSGD5BPROT12 FSGD5BSPID12 | |
| 13 | W | FSGD5BPROT13 FSGD5BSPID13 | | | |

Table 31.157 Peripheral Modules to be Protected and the Corresponding PBG Channel Numbers (14/14)

| PBG group | PBG channel number | Purpose (R or W) | Module to be protected [module name] Register | Guard register | Error status of PBG group |
|-----------|--------------------|------------------|---|------------------------------|--|
| PBG5 #1 | 14 | R | [Flash] (See the Flash Memory User's Manual: Hardware Interface) | FSGD5BPROT14 FSGD5BSPID14 | ERRSLV5CTL ERRSLV5STAT ERRSLV5TYPE |
| | 15 | W | FPMON FASTAT FAEINT FAEASELC FSADDR FEADDR FSTATR FENTRYR FPROTR FSUINITR FLKSTAT FRTSTAT FRTEINT FCMDR FPESTAT FBCCNT FBCSTAT FPSADDR FCPSR FPCKAR FLEMU FLEAD FECCEMON FECCTMD FDMYECC FACI command-issuing area | FSGD5BPROT15 FSGD5BSPID15 | |
| PBG5 #2 | 0 | R | [FLMD] FLMDCNT | FSGD5CPROT00 FSGD5CSPID00 | ERRSLV5CTL ERRSLV5STAT ERRSLV5TYPE |
| | 1 | W | FLMDPCMD FLMDPS | FSGD5CPROT01 FSGD5CSPID01 | |
| | 2 | R | [Flash] OPBT0 | FSGD5CPROT02 FSGD5CSPID02 | |
| | 3 | W | GREG8 TSNREFD PRDNAME _n (n=1 to 4) | FSGD5CPROT03 FSGD5CSPID03 | |
| | 4 | R | [BRAM] BRAMDAT _n (n=0 to 3) | FSGD5CPROT04 FSGD5CSPID04 | |
| | 5 | W | | FSGD5CPROT05 FSGD5CSPID05 | |
| | 6 | | Reserved | | |
| | 7 | | Reserved | | |
| | 8 | | Reserved | | |
| | 9 | | Reserved | | |
| | 10 | | Reserved | | |
| | 11 | | Reserved | | |
| | 12 | | Reserved | | |
| | 13 | | Reserved | | |
| | 14 | | Reserved | | |
| 15 | | Reserved | | | |

Note 1. Address area of Data Flash is protected.

31.4.3.1 List of Registers

Table 31.158 List of Registers (1/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|------------|-----------------------|--|-----|------------------------|-------------|--------------|
| FFC4 C000H | APBFSGDPROT_PDM_ACM_A | P-bus FS Guard Protection Setting Register | R/W | 060D FE1B _H | 8/16/32 | 31.4.3.2 (1) |
| FFC4 C004H | APBFSGDSPID_PDMA_CM_A | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C008H | APBFSGDPROT_PDM_ACM_B | P-bus FS Guard Protection Setting Register | R/W | 060D FE17 _H | 8/16/32 | 31.4.3.2 (2) |
| FFC4 C00CH | APBFSGDSPID_PDMA_CM_B | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C010H | APBFSGDPROT_PDM_ACH_A | P-bus FS Guard Protection Setting Register | R/W | 060D FE1B _H | 8/16/32 | 31.4.3.2 (1) |
| FFC4 C014H | APBFSGDSPID_PDMA_CH_A | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C018H | APBFSGDPROT_PDM_ACH_B | P-bus FS Guard Protection Setting Register | R/W | 060D FE17 _H | 8/16/32 | 31.4.3.2 (2) |
| FFC4 C01CH | APBFSGDSPID_PDMA_CH_B | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C020H | APBFSGDPROT_INTC2_A | P-bus FS Guard Protection Setting Register | R/W | 060D FE1B _H | 8/16/32 | 31.4.3.2 (1) |
| FFC4 C024H | APBFSGDSPID_INTC2_A | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C028H | APBFSGDPROT_INTC2_B | P-bus FS Guard Protection Setting Register | R/W | 060D FE17 _H | 8/16/32 | 31.4.3.2 (2) |
| FFC4 C02CH | APBFSGDSPID_INTC2_B | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C040H | APBFSGDPROT_PBG_A | P-bus FS Guard Protection Setting Register | R/W | 064D FE1B _H | 8/16/32 | 31.4.3.2 (3) |
| FFC4 C044H | APBFSGDSPID_PBG_A | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C048H | APBFSGDPROT_PBG_B | P-bus FS Guard Protection Setting Register | R/W | 064D FE17 _H | 8/16/32 | 31.4.3.2 (4) |
| FFC4 C04CH | APBFSGDSPID_PBG_B | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C050H | APBFSGDPROT_SP1_A | P-bus FS Guard Protection Setting Register | R/W | 060D FE1B _H | 8/16/32 | 31.4.3.2 (1) |
| FFC4 C054H | APBFSGDSPID_SP1_A | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C058H | APBFSGDPROT_SP1_B | P-bus FS Guard Protection Setting Register | R/W | 060D FE17 _H | 8/16/32 | 31.4.3.2 (2) |
| FFC4 C05CH | APBFSGDSPID_SP1_B | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C070H | APBFSGDPROT_SP4_A | P-bus FS Guard Protection Setting Register | R/W | 060D FE1B _H | 8/16/32 | 31.4.3.2 (1) |
| FFC4 C074H | APBFSGDSPID_SP4_A | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C078H | APBFSGDPROT_SP4_B | P-bus FS Guard Protection Setting Register | R/W | 060D FE17 _H | 8/16/32 | 31.4.3.2 (2) |
| FFC4 C07CH | APBFSGDSPID_SP4_B | P-bus FS Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (5) |
| FFC4 C800H | ERRSLVCTL_PFSS0 | ERRSLV Control Register for PFSS P-bus FS Guard Slave0 | W | 0000 0000 _H | 32 | 31.4.3.2 (6) |
| FFC4 C804H | ERRSLVSTAT_PFSS0 | ERRSLV Status Register for PFSS P-bus FS Guard Slave0 | R | 0000 0000 _H | 32 | 31.4.3.2 (7) |
| FFC4 C80CH | ERRSLVTYPE_PFSS0 | ERRSLV Error Transfer Type Register for PFSS P-bus FS Guard Slave0 | R | 0000 0000 _H | 32 | 31.4.3.2 (8) |
| FFC4 C810H | ERRSLVCTL_PFSS1 | ERRSLV Control Register for PFSS P-bus FS Guard Slave1 | W | 0000 0000 _H | 32 | 31.4.3.2 (6) |

Table 31.158 List of Registers (2/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|------------------------|------------------|--|-----|------------------------|-------------|---------------|
| FFC4 C814H | ERRSLVSTAT_PFSS1 | ERRSLV Status Register for PFSS P-bus FS Guard Slave1 | R | 0000 0000 _H | 32 | 31.4.3.2 (7) |
| FFC4 C81CH | ERRSLVTYPE_PFSS1 | ERRSLV Error Transfer Type Register for PFSS P-bus FS Guard Slave1 | R | 0000 0000 _H | 32 | 31.4.3.2 (8) |
| FFC5 A000 _H | FSGD0PROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A004 _H | FSGD0SPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A008 _H | FSGD0PROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A00C _H | FSGD0SPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A020 _H | FSGD0PROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A024 _H | FSGD0SPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A028 _H | FSGD0PROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A02C _H | FSGD0SPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A030 _H | FSGD0PROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A034 _H | FSGD0SPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A038 _H | FSGD0PROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A03C _H | FSGD0SPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A050 _H | FSGD0PROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A054 _H | FSGD0SPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A058 _H | FSGD0PROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC5 A05C _H | FSGD0SPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC5 A100 _H | ERRSLV0CTL | ERRSLV Control Register for P-bus Guard | W | 00 _H | 8 | 31.4.3.2 (11) |
| FFC5 A104 _H | ERRSLV0STAT | ERRSLV Status Register for P-bus Guard | R | 0000 0000 _H | 32 | 31.4.3.2 (12) |
| FFC5 A10C _H | ERRSLV0TYPE | ERRSLV Error Transfer Type Register for P-bus Guard | R | 0000 0000 _H | 32 | 31.4.3.2 (13) |
| FFDDD000 _H | FSGD1APROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD004 _H | FSGD1ASPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD008 _H | FSGD1APROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD00C _H | FSGD1ASPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD010 _H | FSGD1APROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD014 _H | FSGD1ASPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD018 _H | FSGD1APROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD01C _H | FSGD1ASPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (3/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFDDD020 _H | FSGD1APROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD024 _H | FSGD1ASPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD028 _H | FSGD1APROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD02C _H | FSGD1ASPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD030 _H | FSGD1APROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD034 _H | FSGD1ASPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD038 _H | FSGD1APROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD03C _H | FSGD1ASPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD040 _H | FSGD1APROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD044 _H | FSGD1ASPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD048 _H | FSGD1APROT09 | P-bus Guard Protection Setting Register | R/W | 0605FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD04C _H | FSGD1ASPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD050 _H | FSGD1APROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD054 _H | FSGD1ASPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD058 _H | FSGD1APROT11 | P-bus Guard Protection Setting Register | R/W | 0605FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD05C _H | FSGD1ASPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD060 _H | FSGD1APROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD064 _H | FSGD1ASPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD068 _H | FSGD1APROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD06C _H | FSGD1ASPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD070 _H | FSGD1APROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD074 _H | FSGD1ASPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD078 _H | FSGD1APROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD07C _H | FSGD1ASPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD080 _H | FSGD1BPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD084 _H | FSGD1BSPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD088 _H | FSGD1BPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD08C _H | FSGD1BSPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD090 _H | FSGD1BPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD094 _H | FSGD1BSPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (4/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFDDD098 _H | FSGD1BPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD09C _H | FSGD1BSPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0A0 _H | FSGD1BPROT04 | P-bus Guard Protection Setting Register | R/W | 0605FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0A4 _H | FSGD1BSPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0A8 _H | FSGD1BPROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0AC _H | FSGD1BSPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0B0 _H | FSGD1BPROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0B4 _H | FSGD1BSPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0B8 _H | FSGD1BPROT07 | P-bus Guard Protection Setting Register | R/W | 0605FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0BC _H | FSGD1BSPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0C0 _H | FSGD1BPROT08 | P-bus Guard Protection Setting Register | R/W | 0605FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0C4 _H | FSGD1BSPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0C8 _H | FSGD1BPROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0CC _H | FSGD1BSPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0D0 _H | FSGD1BPROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0D4 _H | FSGD1BSPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0D8 _H | FSGD1BPROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0DC _H | FSGD1BSPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0E0 _H | FSGD1BPROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0E4 _H | FSGD1BSPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0E8 _H | FSGD1BPROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0EC _H | FSGD1BSPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0F0 _H | FSGD1BPROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0F4 _H | FSGD1BSPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD0F8 _H | FSGD1BPROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD0FC _H | FSGD1BSPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD100 _H | FSGD1CPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD104 _H | FSGD1CSPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD108 _H | FSGD1CPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD10C _H | FSGD1CSPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (5/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFDDD110 _H | FSGD1CPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD114 _H | FSGD1CSPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD118 _H | FSGD1CPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD11C _H | FSGD1CSPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD120 _H | FSGD1CPROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD124 _H | FSGD1CSPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD128 _H | FSGD1CPROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD12C _H | FSGD1CSPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD130 _H | FSGD1CPROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD134 _H | FSGD1CSPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD138 _H | FSGD1CPROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD13C _H | FSGD1CSPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD140 _H | FSGD1CPROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD144 _H | FSGD1CSPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD148 _H | FSGD1CPROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD14C _H | FSGD1CSPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD150 _H | FSGD1CPROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD154 _H | FSGD1CSPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD158 _H | FSGD1CPROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD15C _H | FSGD1CSPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD160 _H | FSGD1CPROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD164 _H | FSGD1CSPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD168 _H | FSGD1CPROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD16C _H | FSGD1CSPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD170 _H | FSGD1CPROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD174 _H | FSGD1CSPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD178 _H | FSGD1CPROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD17C _H | FSGD1CSPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD180 _H | FSGD1DPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD184 _H | FSGD1DSPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (6/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFDDD188 _H | FSGD1DPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD18C _H | FSGD1DSPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD190 _H | FSGD1DPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD194 _H | FSGD1DSPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD198 _H | FSGD1DPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD19C _H | FSGD1DSPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1A0 _H | FSGD1DPROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1A4 _H | FSGD1DSPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1A8 _H | FSGD1DPROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1AC _H | FSGD1DSPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1B0 _H | FSGD1DPROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1B4 _H | FSGD1DSPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1B8 _H | FSGD1DPROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1BC _H | FSGD1DSPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1C0 _H | FSGD1DPROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1C4 _H | FSGD1DSPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1C8 _H | FSGD1DPROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1CC _H | FSGD1DSPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1D0 _H | FSGD1DPROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1D4 _H | FSGD1DSPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1D8 _H | FSGD1DPROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1DC _H | FSGD1DSPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1E0 _H | FSGD1DPROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1E4 _H | FSGD1DSPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1E8 _H | FSGD1DPROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1EC _H | FSGD1DSPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1F0 _H | FSGD1DPROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1F4 _H | FSGD1DSPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD1F8 _H | FSGD1DPROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD1FC _H | FSGD1DSPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (7/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFDDD200 _H | FSGD1EPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD204 _H | FSGD1ESPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD208 _H | FSGD1EPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD20C _H | FSGD1ESPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD210 _H | FSGD1EPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD214 _H | FSGD1ESPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD218 _H | FSGD1EPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD21C _H | FSGD1ESPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD220 _H | FSGD1EPROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD224 _H | FSGD1ESPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD228 _H | FSGD1EPROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD22C _H | FSGD1ESPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD230 _H | FSGD1EPROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD234 _H | FSGD1ESPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD238 _H | FSGD1EPROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD23C _H | FSGD1ESPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD240 _H | FSGD1EPROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD244 _H | FSGD1ESPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD248 _H | FSGD1EPROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD24C _H | FSGD1ESPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD250 _H | FSGD1EPROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD254 _H | FSGD1ESPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD258 _H | FSGD1EPROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD25C _H | FSGD1ESPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD260 _H | FSGD1EPROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD264 _H | FSGD1ESPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD268 _H | FSGD1EPROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD26C _H | FSGD1ESPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD270 _H | FSGD1EPROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD274 _H | FSGD1ESPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (8/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFDDD278 _H | FSGD1EPROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFDDD27C _H | FSGD1ESPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFDDD400 _H | ERRSLV1CTL | ERRSLV Control Register for P-bus Guard | W | 00 _H | 8 | 31.4.3.2 (11) |
| FFDDD404 _H | ERRSLV1STAT | ERRSLV Status Register for P-bus Guard | R | 00000000 _H | 32 | 31.4.3.2 (12) |
| FFDDD40C _H | ERRSLV1TYPE | ERRSLV Error Transfer Type Register for P-bus Guard | R | 00000000 _H | 32 | 31.4.3.2 (13) |
| FFF94000 _H | FSGD2APROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94004 _H | FSGD2ASPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94008 _H | FSGD2APROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9400C _H | FSGD2ASPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94010 _H | FSGD2APROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94014 _H | FSGD2ASPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94018 _H | FSGD2APROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9401C _H | FSGD2ASPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94020 _H | FSGD2APROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94024 _H | FSGD2ASPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94028 _H | FSGD2APROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9402C _H | FSGD2ASPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94030 _H | FSGD2APROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94034 _H | FSGD2ASPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94038 _H | FSGD2APROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9403C _H | FSGD2ASPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94040 _H | FSGD2APROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94044 _H | FSGD2ASPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94048 _H | FSGD2APROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9404C _H | FSGD2ASPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94050 _H | FSGD2APROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94054 _H | FSGD2ASPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94058 _H | FSGD2APROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9405C _H | FSGD2ASPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94060 _H | FSGD2APROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |

Table 31.158 List of Registers (9/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFF94064 _H | FSGD2ASPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94068 _H | FSGD2APROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9406C _H | FSGD2ASPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94070 _H | FSGD2APROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94074 _H | FSGD2ASPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94078 _H | FSGD2APROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9407C _H | FSGD2ASPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94080 _H | FSGD2BPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94084 _H | FSGD2BSPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94088 _H | FSGD2BPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9408C _H | FSGD2BSPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94090 _H | FSGD2BPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94094 _H | FSGD2BSPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94098 _H | FSGD2BPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9409C _H | FSGD2BSPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940A0 _H | FSGD2BPROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940A4 _H | FSGD2BSPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940A8 _H | FSGD2BPROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940AC _H | FSGD2BSPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940B0 _H | FSGD2BPROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940B4 _H | FSGD2BSPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940B8 _H | FSGD2BPROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940BC _H | FSGD2BSPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940C0 _H | FSGD2BPROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940C4 _H | FSGD2BSPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940C8 _H | FSGD2BPROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940CC _H | FSGD2BSPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940D0 _H | FSGD2BPROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940D4 _H | FSGD2BSPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940D8 _H | FSGD2BPROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |

Table 31.158 List of Registers (10/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFF940DC _H | FSGD2BSPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940E0 _H | FSGD2BPROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940E4 _H | FSGD2BSPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF940E8 _H | FSGD2BPROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF940EC _H | FSGD2BSPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94100 _H | FSGD2CPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94104 _H | FSGD2CSPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94108 _H | FSGD2CPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9410C _H | FSGD2CSPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94110 _H | FSGD2CPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94114 _H | FSGD2CSPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94118 _H | FSGD2CPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9411C _H | FSGD2CSPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94120 _H | FSGD2CPROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94124 _H | FSGD2CSPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94128 _H | FSGD2CPROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9412C _H | FSGD2CSPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94130 _H | FSGD2CPROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94134 _H | FSGD2CSPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94138 _H | FSGD2CPROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9413C _H | FSGD2CSPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94140 _H | FSGD2CPROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94144 _H | FSGD2CSPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94148 _H | FSGD2CPROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9414C _H | FSGD2CSPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94150 _H | FSGD2CPROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFF94154 _H | FSGD2CSPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94158 _H | FSGD2CPROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFF9415C _H | FSGD2CSPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFF94400 _H | ERRSLV2CTL | ERRSLV Control Register for P-bus Guard | W | 00 _H | 8 | 31.4.3.2 (11) |

Table 31.158 List of Registers (11/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFF94404 _H | ERRSLV2STAT | ERRSLV Status Register for P-bus Guard | R | 00000000 _H | 32 | 31.4.3.2 (12) |
| FFF9440C _H | ERRSLV2TYPE | ERRSLV Error Transfer Type Register for P-bus Guard | R | 00000000 _H | 32 | 31.4.3.2 (13) |
| FFC40000 _H | FSGD5APROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40004 _H | FSGD5ASPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40008 _H | FSGD5APROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4000C _H | FSGD5ASPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40010 _H | FSGD5APROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40014 _H | FSGD5ASPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40018 _H | FSGD5APROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4001C _H | FSGD5ASPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40020 _H | FSGD5APROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40024 _H | FSGD5ASPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40028 _H | FSGD5APROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4002C _H | FSGD5ASPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40030 _H | FSGD5APROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40034 _H | FSGD5ASPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40038 _H | FSGD5APROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4003C _H | FSGD5ASPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40040 _H | FSGD5APROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40044 _H | FSGD5ASPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40048 _H | FSGD5APROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4004C _H | FSGD5ASPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40050 _H | FSGD5APROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40054 _H | FSGD5ASPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40058 _H | FSGD5APROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4005C _H | FSGD5ASPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40060 _H | FSGD5APROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40064 _H | FSGD5ASPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40068 _H | FSGD5APROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4006C _H | FSGD5ASPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (12/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFC40070 _H | FSGD5APROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40074 _H | FSGD5ASPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40078 _H | FSGD5APROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4007C _H | FSGD5ASPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40080 _H | FSGD5BPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40084 _H | FSGD5BSPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40088 _H | FSGD5BPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4008C _H | FSGD5BSPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40090 _H | FSGD5BPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40094 _H | FSGD5BSPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40098 _H | FSGD5BPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4009C _H | FSGD5BSPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400B0 _H | FSGD5BPROT06 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400B4 _H | FSGD5BSPID06 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400B8 _H | FSGD5BPROT07 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400BC _H | FSGD5BSPID07 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400C0 _H | FSGD5BPROT08 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400C4 _H | FSGD5BSPID08 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400C8 _H | FSGD5BPROT09 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400CC _H | FSGD5BSPID09 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400D0 _H | FSGD5BPROT10 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400D4 _H | FSGD5BSPID10 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400D8 _H | FSGD5BPROT11 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400DC _H | FSGD5BSPID11 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400E0 _H | FSGD5BPROT12 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400E4 _H | FSGD5BSPID12 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400E8 _H | FSGD5BPROT13 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400EC _H | FSGD5BSPID13 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC400F0 _H | FSGD5BPROT14 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400F4 _H | FSGD5BSPID14 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |

Table 31.158 List of Registers (13/13)

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|---|-----|------------------------|-------------|---------------|
| FFC400F8 _H | FSGD5BPROT15 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC400FC _H | FSGD5BSPID15 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40100 _H | FSGD5CPROT00 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40104 _H | FSGD5CSPID00 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40108 _H | FSGD5CPROT01 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4010C _H | FSGD5CSPID01 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40110 _H | FSGD5CPROT02 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40114 _H | FSGD5CSPID02 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40118 _H | FSGD5CPROT03 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4011C _H | FSGD5CSPID03 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40120 _H | FSGD5CPROT04 | P-bus Guard Protection Setting Register | R/W | 0605 FE1B _H | 8/16/32 | 31.4.3.2 (9) |
| FFC40124 _H | FSGD5CSPID04 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40128 _H | FSGD5CPROT05 | P-bus Guard Protection Setting Register | R/W | 0605 FE17 _H | 8/16/32 | 31.4.3.2 (9) |
| FFC4012C _H | FSGD5CSPID05 | P-bus Guard SPID Setting Register | R/W | FFFF FFFF _H | 32 | 31.4.3.2 (10) |
| FFC40400 _H | ERRSLV5CTL | ERRSLV Control Register for P-bus Guard | W | 00 _H | 8 | 31.4.3.2 (11) |
| FFC40404 _H | ERRSLV5STAT | ERRSLV Status Register for P-bus Guard | R | 00000000 _H | 32 | 31.4.3.2 (12) |
| FFC4040C _H | ERRSLV5TYPE | ERRSLV Error Transfer Type Register for P-bus Guard | R | 00000000 _H | 32 | 31.4.3.2 (13) |

31.4.3.2 Details of Registers

(1) APBFSGDPROT_PDMACM_A/PDMACH_A/INTC2_A/SPn_A — P-bus FS Guard Protection Setting Register (n = 1, 4)

These registers are for setting P-bus guard in read cycles.

Access: These registers can be read/written in 32/16/8-bit units.

Address: APBFSGDPROT_PDMACM_A: FFC4 C000_H
 APBFSGDPROT_PDMACH_A: FFC4 C010_H
 APBFSGDPROT_INTC2_A: FFC4 C020_H
 APBFSGDPROT_SP1_A: FFC4 C050_H
 APBFSGDPROT_SP4_A: FFC4 C070_H

Value after reset: 060D FE1B_H

| | | | | | | | | | | | | | | | | |
|-------------------|------|----|----|----|----|----|----|----|----|----|-------|----|----|-------|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | LOCK | — | — | — | — | — | — | — | — | — | PEID4 | — | — | PEID1 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.159 APBFSGDPROT_PDMACM_A/PDMACH_A/INTC2_A/SPn_A Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | LOCK | Lock bit 0: Registers can be-written 1: Any further writes to APBFSGDPROT_x and APBFSGDSPID_x are ignored. This bit can only be cleared by reset. |
| 30 to 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | PEID4 | Access with PEID4 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 20 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PEID1 | Access with PEID1 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 17 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(2) APBFSGDPROT_PDMACM_B/PDMACH_B/INTC2_B/SPn_B — P-bus FS Guard Protection Setting Register (n = 1, 4)

These registers are for setting P-bus guard in write cycles.

Access: These registers can be read/written in 32/16/8-bit units.

Address: APBFSGDPROT_PDMACM_B: FFC4 C008_H
 APBFSGDPROT_PDMACH_B: FFC4 C018_H
 APBFSGDPROT_INTC2_B: FFC4 C028_H
 APBFSGDPROT_SP1_B: FFC4 C058_H
 APBFSGDPROT_SP4_B: FFC4 C078_H

Value after reset: 060D FE17_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|----|----|----|----|----|----|----|----|----|-------|----|----|-------|----|----|
| | LOCK | — | — | — | — | — | — | — | — | — | PEID4 | — | — | PEID1 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.160 APBFSGDPROT_PDMACM_B/PDMACH_B/INTC2_B/SPn_B Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | LOCK | Lock bit 0: Registers can be-written 1: Any further writes to APBFSGDPROT_x and APBFSGDSPID_x are ignored. This bit can only be cleared by reset |
| 30 to 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | PEID4 | Access with PEID4 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 20 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PEID1 | Access with PEID1 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 17 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(3) APBFSGDPROT_PBG_A — P-bus FS Guard Protection Setting Register

This register is for setting P-bus guard in read cycles.

Access: This register can be read/written in 32/16/8-bit units.

Address: FFC4 C040_H

Value after reset: 064D FE1B_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|----|----|----|----|----|----|----|----|----|-------|----|----|-------|----|----|
| | LOCK | — | — | — | — | — | — | — | — | — | PEID4 | — | — | PEID1 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.161 APBFSGDPROT_PBG_A Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | LOCK | Lock bit 0: Registers can be-written 1: Any further writes to APBFSGDPROT_PBG_A and APBFSGDSPID_PBG_A are ignored. This bit can only be cleared by reset |
| 30 to 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | PEID4 | Access with PEID4 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 20 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PEID1 | Access with PEID1 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 17 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(4) APBFSGDPROT_PBG_B — P-bus FS Guard Protection Setting Register

This register is for setting P-bus guard in write cycles.

Access: This register can be read/written in 32/16/8-bit units.

Address: FFC4 C048_H

Value after reset: 064D FE17_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|----|----|----|----|----|----|----|----|----|-------|----|----|-------|----|----|
| | LOCK | — | — | — | — | — | — | — | — | — | PEID4 | — | — | PEID1 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.162 APBFSGDPROT_PBG_B Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | LOCK | Lock bit 0: Registers can be-written 1: Any further writes to APBFSGDPROT_PBG_B and APBFSGDSPID_PBG_B are ignored. This bit can only be cleared by reset |
| 30 to 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | PEID4 | Access with PEID4 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 20 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PEID1 | Access with PEID1 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEIDn is not allowed. 1: Access with PEIDn is allowed. |
| 17 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(5) APBFSGDSPID_PDMACM_x/PDMACH_x/INTC2_x/PBG_x/SPy_x — P-bus FS Guard SPID Setting Register (x = A or B, y = 1, 4)

“A” is for setting read cycles, “B” is for setting write cycles.

Access: These registers can be read/written in 32-bit units.

Address: APBFSGDSPID_INTC2_A: FFC4 C024_H
 APBFSGDSPID_INTC2_B: FFC4 C02C_H
 APBFSGDSPID_PDMACM_A: FFC4 C004_H
 APBFSGDSPID_PDMACM_B: FFC4 C00C_H
 APBFSGDSPID_PDMACH_A: FFC4 C014_H
 APBFSGDSPID_PDMACH_B: FFC4 C01C_H
 APBFSGDSPID_PBG_A: FFC4 C044_H
 APBFSGDSPID_PBG_B: FFC4 C04C_H
 APBFSGDSPID_SP1_A: FFC4 C054_H
 APBFSGDSPID_SP1_B: FFC4 C05C_H
 APBFSGDSPID_SP4_A: FFC4 C074_H
 APBFSGDSPID_SP4_B: FFC4 C07C_H

Value after reset: FFFF FFFF_H

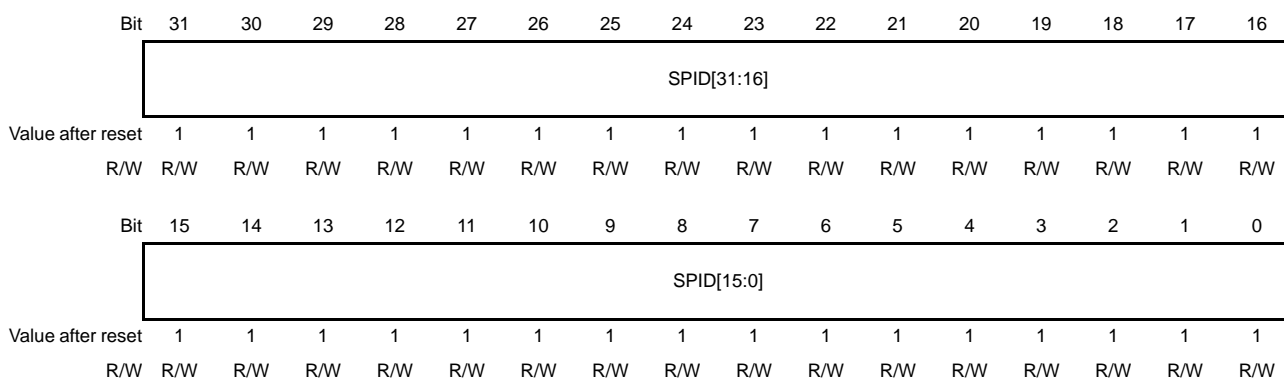


Table 31.163 APBFSGDSPID_PDMACM_x/PDMACH_x/INTC2_x/PBG_x/SPy_x Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 0 | SPID[31:0] | Access with SPID SPID is a bit list where each bit represents one SPID value. Setting more than one bit allows to enable more than one SPID value at a time. E.g. setting SPID to “00000110” allows access with SPID = 1 and SPID = 2. 0: Access with SPIDn is not allowed. 1: Access with SPIDn is allowed. |

(6) ERRSLVCTL_PFSS0/1 — ERRSLV Control Register for PFSS P-bus FS Guard Slave0/1

These registers are for setting P-bus guard in read cycles.

Access: These registers can be written only in 32-bit units.

Address: ERRSLVCTL_PFSS0: FFC4 C800_H
ERRSLVCTL_PFSS1: FFC4 C810_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLRO | CLRE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | W | W |

Table 31.164 ERRSLVCTL_PFSS0/1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When writing, write the value after reset. |
| 1 | CLRO | Clear the OVF bit of ERRSLVSTAT_PFSS0/1 by writing “1” to this bit. |
| 0 | CLRE | Clear the ERR bit of ERRSLVSTAT_PFSS0/1 by writing “1” to this bit. |

(7) ERRSLVSTAT_PFSS0/1 — ERRSLV Status Register for PFSS P-bus FS Guard Slave0/1

Access: These registers can be read in 32-bit units.

Address: ERRSLVSTAT_PFSS0: FFC4 C804_H
ERRSLVSTAT_PFSS1: FFC4 C814_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVF | ERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.165 ERRSLVSTAT_PFSS0/1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | OVF | Error Overflow Status Flag 0: No Overflow 1: Error Overflow |
| 0 | ERR | Error Status Flag 0: No Error 1: Error |

(8) ERRSLVTYPE_PFSS0/1 — ERRSLV Error Transfer Type Register for PFSS P-bus FS Guard Slave0/1

Access: These registers can be read in 32-bit units.

Address: ERRSLVTYPE_PFSS0: FFC4 C80C_H
ERRSLVTYPE_PFSS1: FFC4 C81C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | SPID[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PEID[2:0] | | | — | — | — | — | — | — | — | — | — | — | — | — | WRITE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.166 ERRSLVTYPE_PFSS0/1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. |
| 20 to 16 | SPID[4:0] | P-bus guard SPID error has occurred |
| 15 to 13 | PEID[2:0] | P-bus guard PEID error has occurred |
| 12 to 1 | Reserved | These bits are undefined. |
| 0 | WRITE | Access type of P-bus Guard error 0: read access 1: write access |

(9) xxxPROTx — P-bus Guard Protection Setting Register

Access: These registers can be read/written in 32/16/8-bit units.

Address: see **Table 31.158, List of Registers** (P-bus Guard)

Value after reset: see **Table 31.158, List of Registers** (P-bus Guard)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|----|----|----|----|----|----|----|----|----|-------|----|----|-------|----|----|
| | LOCK | — | — | — | — | — | — | — | — | — | PEID4 | — | — | PEID1 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.167 xxxPROTx Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | LOCK | Lock bit 0: Registers can be-written 1: Any further writes to xxxPROTx and xxxSPIDx are ignored. This bit can only be cleared by reset |
| 30 to 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | PEID4 | Access with PEID4 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEID n is not allowed. 1: Access with PEID n is allowed. |
| 20 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PEID1 | Access with PEID1 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEID n is not allowed. 1: Access with PEID n is allowed. |
| 17 to 4 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 3 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. Please refer to the Table 31.158 for the read values of these bits. |

(10) xxxSPIDx — P-bus Guard SPID Setting Register

Access: These registers can be read/written in 32-bit units.

Address: see Table 31.158, List of Registers (P-bus Guard)

Value after reset: FFFF FFFF_H

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | SPID[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | SPID[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.168 xxxSPIDx Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|--|
| 31 to 0 | SPID[31:0] | <p>Access with SPID</p> <p>SPID is a bit list where each bit represents one SPID value. Setting more than one bit allows to enable more than one SPID value at a time. E.g. setting SPID to "00000110" allows access with SPID = 1 and SPID = 2.</p> <p>0: Access with SPID n is not allowed.</p> <p>1: Access with SPID n is allowed.</p> |

(11) ERRSLVxCTL — ERRSLV Control Register for P-bus Guard

Access: These registers can be written only in 8-bit units.

Address: see Table 31.158, List of Registers (P-bus Guard)

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | CLRO | CLRE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | W | W |

Table 31.169 ERRSLVxCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 2 | Reserved | When writing, write the value after reset. |
| 1 | CLRO | Clear the OVF bit of ERRSLVxSTAT by writing “1” to this bit. |
| 0 | CLRE | Clear the ERR bit of ERRSLVxSTAT by writing “1” to this bit. |

(12) ERRSLVxSTAT — ERRSLV Status Register for P-bus Guard

Access: These registers can be read in 32-bit units.

Address: see Table 31.158, List of Registers (P-bus Guard)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVF | ERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.170 ERRSLV1STAT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | OVF | Error Overflow Status Flag 0: No Overflow 1: Error Overflow |
| 0 | ERR | Error Status Flag 0: No Error 1: Error |

(13) ERRSLVxTYPE — ERRSLV Error Transfer Type Register for P-bus Guard

Access: These registers can be read in 32-bit units.

Address: see Table 31.158, List of Registers (P-bus Guard)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | SPID[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PEID[2:0] | | | — | — | — | — | — | — | — | — | — | — | — | — | WRITE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.171 ERRSLVxTYPE Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. |
| 20 to 16 | SPID[4:0] | P-bus guard SPID error has occurred |
| 15 to 13 | PEID[2:0] | P-bus guard PEID error has occurred |
| 12 to 1 | Reserved | These bits are undefined. |
| 0 | WRITE | Access type of P-bus Guard error 0: read access 1: write access |

31.4.4 HBG

H-bus guard protects against read access and write access to the peripheral circuit on H-bus. If H-bus guard detects illegal access, guard error notification is sent to ECM.

Table 31.172 List of Registers

| Address | Symbol | Register Name | R/W | Value after reset | Access Size | Refer to |
|-----------------------|--------------|------------------------------------|-----|------------------------|-------------|--------------|
| FFFA0000 _H | FSGDF0PROT00 | FlexRay0 H-bus Guard for Read | R/W | 0605 FE1B _H | 8/16/32 | 31.4.4.1 (1) |
| FFFA0004 _H | FSGDF0SPID00 | FlexRay0 H-bus Guard for Read | R/W | FFFF FFFF _H | 32 | 31.4.4.1 (3) |
| FFFA0008 _H | FSGDF0PROT01 | FlexRay0 H-bus Guard for Write | R/W | 0605 FE17 _H | 8/16/32 | 31.4.4.1 (2) |
| FFFA000C _H | FSGDF0SPID01 | FlexRay0 H-bus Guard for Write | R/W | FFFF FFFF _H | 32 | 31.4.4.1 (3) |
| FFFA0010 _H | ERRSLVF0CTL | FlexRay0 H-bus Guard Error slave | W | 00 _H | 8 | 31.4.4.1 (4) |
| FFFA0014 _H | ERRSLVF0STAT | FlexRay0 H-bus Guard Error slave | R | 0000 0000 _H | 32 | 31.4.4.1 (5) |
| FFFA001C _H | ERRSLVF0TYPE | FlexRay0 H-bus Guard Error slave | R | 0000 0000 _H | 32 | 31.4.4.1 (6) |
| FFFA1000 _H | HSSPIDRG0 | FlexRay0 H-bus master SPID setting | R/W | 0000 0000 _H | 32 | 31.4.4.1 (7) |

31.4.4.1 Details of Registers

(1) FSGDxxPROT00

Access: These registers can be read/written in 32/16/8-bit units.

Address: see Table 31.172, List of Registers (H-bus Guard)

Value after reset: 0605 FE1B_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|----|----|----|----|----|----|----|----|----|-------|----|----|-------|----|----|
| | LOCK | — | — | — | — | — | — | — | — | — | PEID4 | — | — | PEID1 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.173 FSGDxxPROTnn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | LOCK | Lock bit 0: Registers can be-written 1: Any further writes to FSGDxxPROTn and FSGDxxSPIDn are ignored. This bit can only be cleared by reset |
| 30 to 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | PEID4 | Access with PEID4 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEID n is not allowed. 1: Access with PEID n is allowed. |
| 20 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PEID1 | Access with PEID1 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEID n is not allowed. 1: Access with PEID n is allowed. |
| 17 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(2) FSGDxxPROT01

Access: These registers can be read/written in 32/16/8-bit units.

Address: see Table 31.172, List of Registers (H-bus Guard)

Value after reset: 0605 FE17_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------|----|----|----|----|----|----|----|----|----|-------|----|----|-------|----|----|
| | LOCK | — | — | — | — | — | — | — | — | — | PEID4 | — | — | PEID1 | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| R/W | R/W | R | R | R | R | R | R | R | R | R | R/W | R | R | R/W | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.174 FSGDxxPROTnn Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 | LOCK | Lock bit 0: Registers can be-written 1: Any further writes to FSGDxxPROTn and FSGDxxSPIDn are ignored. This bit can only be cleared by reset |
| 30 to 22 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 21 | PEID4 | Access with PEID4 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEID n is not allowed. 1: Access with PEID n is allowed. |
| 20 to 19 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 18 | PEID1 | Access with PEID1 PEID is a bit list where each bit represents one PEID value. Setting more than one bit allows to enable more than one virtual Processor ID value at a time. 0: Access with PEID n is not allowed. 1: Access with PEID n is allowed. |
| 17 to 0 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |

(3) FSGDxxSPIDnn

Access: These registers can be read/written in 32-bit units.

Address: see **Table 31.172, List of Registers** (H-bus Guard)

Value after reset: FFFF FFFF_H

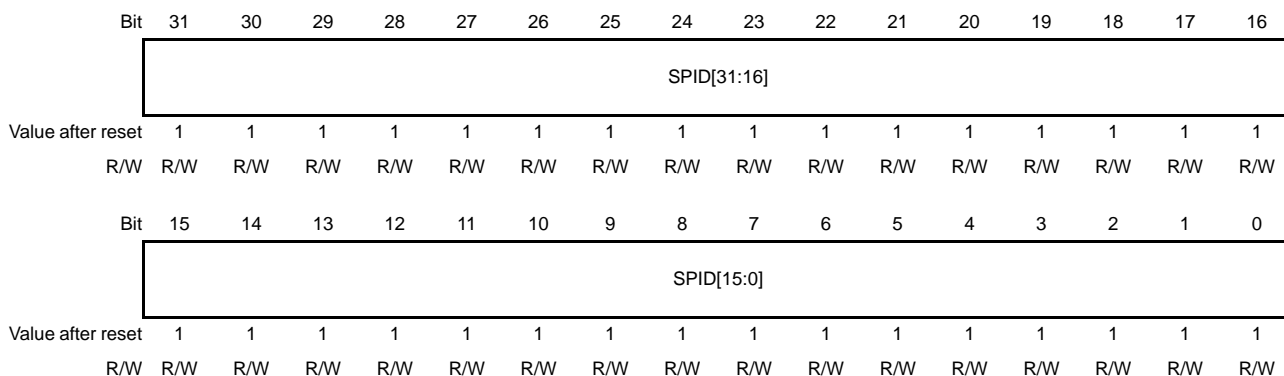


Table 31.175 SGDxxSPIDnn Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 31 to 0 | SPID[31:0] | Access with SPID SPID is a bit list where each bit represents one SPID value. Setting more than one bit allows to enable more than one SPID value at a time. E.g. setting SPID to "00000110" allows access with SPID = 1 and SPID = 2. 0: Access with SPID n is not allowed. 1: Access with SPID n is allowed. |

(4) ERRSLVxxCTL

Access: These registers can be written in 8-bit units.

Address: see Table 31.172, List of Registers (H-bus Guard)

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | CLRO | CLRE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | W | W |

Table 31.176 ERRSLVxxCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 2 | Reserved | When writing, write the value after reset. |
| 1 | CLRO | 0: Ignored 1: Clears the OVF bit of ERRSLVxxSTAT Read value is always "0" |
| 0 | CLRE | 0: Ignored 1: Clears the ERR bit of ERRSLVxxSTAT Read value is always "0" |

(5) ERRSLVxxSTAT

Access: These registers can be read in 32-bit units.

Address: see Table 31.172, List of Registers (H-bus Guard)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OVF | ERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.177 ERRSLVxxSTAT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1 | OVF | Error Overflow Status Flag 0: No Overflow 1: Error Overflow |
| 0 | ERR | Error Status Flag 0: No Error 1: Error |

(6) ERRSLVxxTYPE

Access: These registers can be read in 32-bit units.

Address: see Table 31.172, List of Registers (H-bus Guard)

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|-------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| | — | | | | | | | | | | | SPID[4:0] | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | PEID[2:0] | | | — | | | | | | | | | | | | | WRITE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | |

Table 31.178 ERRSLVxxTYPE Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 21 | Reserved | When read, the value after reset is read. |
| 20 to 16 | SPID[4:0] | H-bus guard SPID error has occurred |
| 15 to 13 | PEID[2:0] | H-bus guard PEID error has occurred |
| 12 to 10 | Reserved | These bits are undefined. |
| 9, 8 | Reserved | When read, the value after reset is read. |
| 7, 6 | Reserved | These bits are undefined. |
| 5, 4 | Reserved | When read, the value after reset is read. |
| 3 to 1 | Reserved | These bits are undefined. |
| 0 | WRITE | Access type of H-bus Guard error 0: read access 1: write access |

(7) HSSPIDRG0

This register is setting SPID for H-Bus master IP (FlexRay). The H-Bus master IP can only use a value of SPID between 2 and 31.

Access: These registers can be read/written in 32-bit units.

Address: HSSPIDRG0: FFFA 1000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|-----------|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | SPID[4:0] | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W |

Table 31.179 HSSPIDRGn Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 5 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 4 to 0 | SPID[4:0] | System Protection ID. In case that this value is "0" or "1", SPID of the master is "2" instead of register value. |

Table 31.180 Target IP

| Register Symbol | target IP |
|-----------------|-----------|
| HSSPIDRG0 | FlexRay0 |

31.5 Clock Monitor (CLMA)

This product implements multiple clock monitors that monitor internal clocks related to the safety of the device. Each clock monitor detects clock failure by using two kinds of clocks: the “monitored clock” and the “sampling clock”. The “monitored clock” is the target of monitoring and the “sampling clock” is used as a reference to check the frequency of the “monitored clock”.

31.5.1 Features

- Implements four clock monitor modules.
The individual CLMA units are identified by the index “n”(n = 0 to 3).
- Monitors to see if the frequency of the clock to be monitored is within the specified range based on the sampling clock. Issues an error notice to the ECM upon detection of the abnormal state of the clock.
- Supports self-diagnosis function.

31.5.1.1 Clock Monitor Configurations

Table 31.181 Clock Monitor Configuration

| Clock Monitor | Monitored Clock (CLMATMON) | Sampling Clock (CLMATSMPL) | Status After Reset | Output to |
|---------------|--|----------------------------|--------------------|---|
| CLMA0 | Main OSC | CLK_IOSC / 2 (4 MHz) | Disabled | ECM upper-limit error : Error factor #8 lower-limit error : Error factor #9 |
| CLMA1 | CLK_LSB / 2 (20MHz) | Main OSC / 8 | Disabled | ECM upper-limit error : Error factor #12 lower-limit error : Error factor #13 |
| CLMA2 | Window Watch Dog count clock* ¹ | Main OSC / 256 | Disabled | ECM upper-limit error : Error factor #10 lower-limit error : Error factor #11 |
| CLMA3 | Clock CPU (160 MHz) | Main OSC / 4 | Disabled | ECM upper-limit error : Error factor #14 lower-limit error : Error factor #15 |

Note 1. The monitored clock of CLMA2 is selected by the setting of OPWDMDS.

- OPWDMDS
 - 0: WDTACLKI = CLK_IOSC (8 MHz)
 - 1: WDTACLKI = CLK_IOSC / 32 (250 kHz)

31.5.1.2 Block Diagram

Figure 31.7 shows block diagram of each clock monitor A.

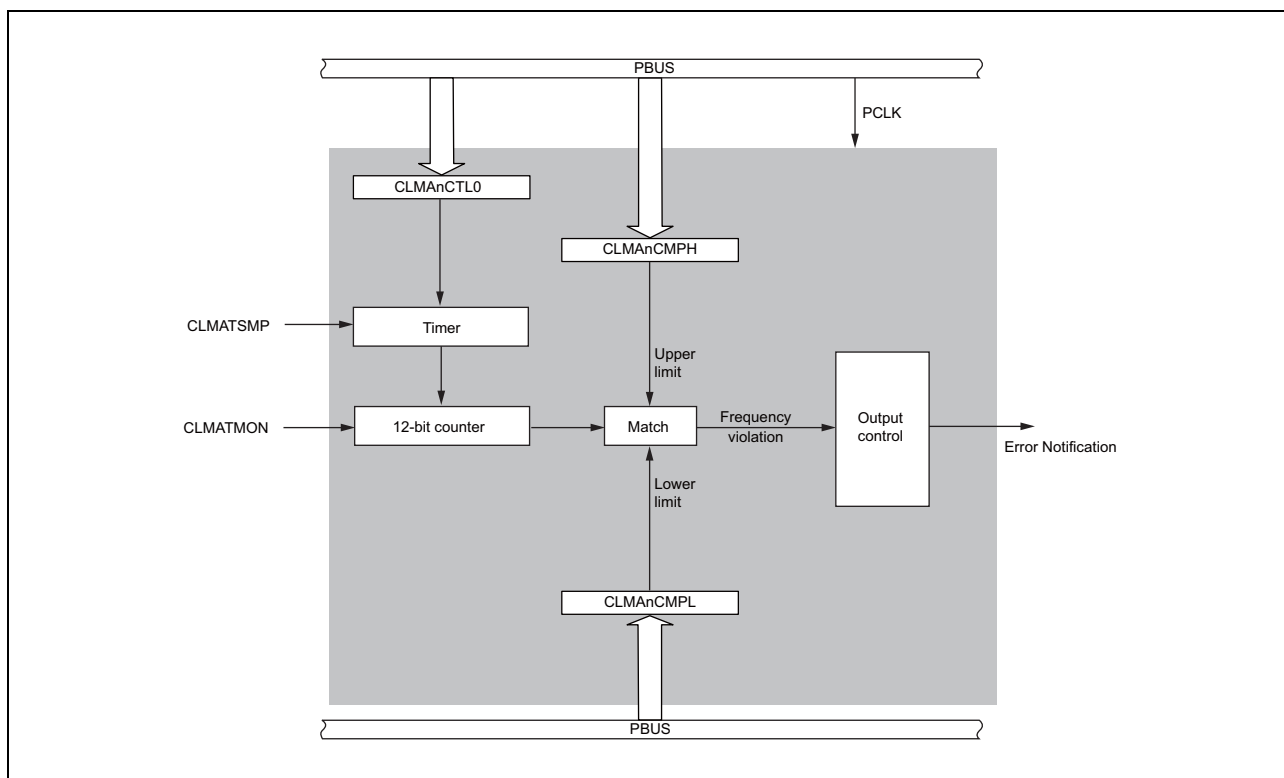


Figure 31.7 Block Diagram of Each Clock Monitor A

31.5.1.3 External Input/Output Pins

None.

31.5.2 Functional Description

The Clock Monitor CLMAN is used to verify whether the frequency of a clock (CLMATMON) is within the specified range.

31.5.2.1 Detection of Abnormal Clock Frequencies

- CLMAN counts the number of rising edges of the monitored clock signal within 16 sampling clock cycles, and compares the count with the specified thresholds.
 - The lower threshold is specified using the CLMANCMPL[11:0] bits in CLMANCMPL register.
 - The upper threshold is specified using the CLMANCMPH[11:0] bits in CLMANCMPH register.
- When the monitored clock frequency is so low*¹ that the count falls below the value set in CLMANCMPL[11:0] in the CLMANCMPL register, CLMAN notifies the ECM of the abnormal clock. CLMAN also notifies the ECM of the abnormal clock when the clock frequency is so high that the count exceeds the value set in CLMANCMPH[11:0] in the CLMANCMPH register.

Note 1. There is a case that the abnormal state is not detected when the monitor clock completely stops.

Note that even if the frequency of the monitored clock fluctuates during the sampling period, an error is not notified as long as the number of detected edges falls within the specified range.

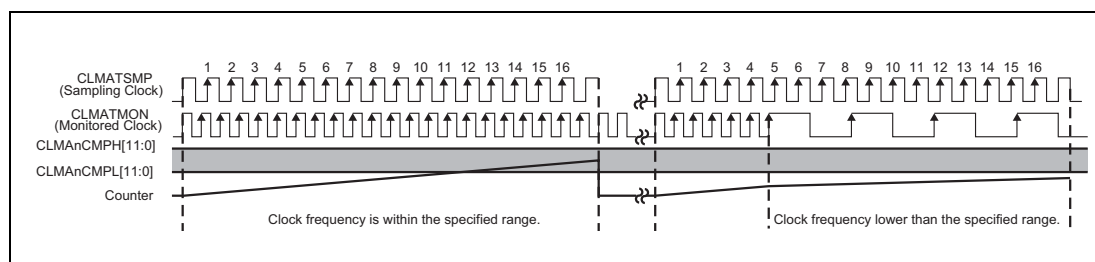


Figure 31.8 Operation when Monitored Clock Frequency is Lower than Lower Threshold

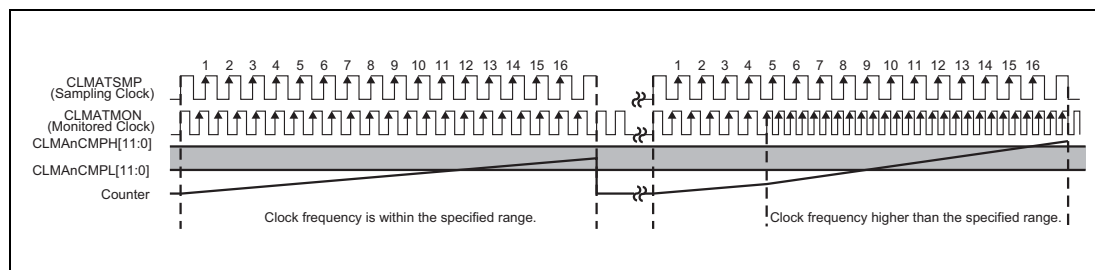


Figure 31.9 Operation when Monitored Clock Frequency is Higher than Higher Threshold

Calculation of thresholds

For the compare registers CLMAnCMPL and CLMAnCMPH, specify the minimum and maximum number of CLMATMON clock cycles to occur within 16 cycles of the sampling clock CLMATSMP that defines the normal range of CLMATMON.

The number of CLMATMON clock cycles to occur within 16 cycles of CLMATSMP is denoted by N.

$$\frac{16}{f_{\text{CLMATSMP}}} = \frac{N}{f_{\text{CLMATMON}}}$$

$$N = \frac{f_{\text{CLMATMON}}}{f_{\text{CLMATSMP}}} \times 16$$

Considering the allowed frequency deviations of CLMATMON and CLMATSMP, the threshold values can be calculated by the following formulas:

$$\begin{aligned} \text{Lower threshold} &= N_{\min} \\ &= \frac{f_{\text{CLMATMON}(\min)}}{f_{\text{CLMATSMP}(\max)}} \times 16 - 1 \end{aligned}$$

$$\begin{aligned} \text{Upper threshold} &= N_{\max} \\ &= \frac{f_{\text{CLMATMON}(\max)}}{f_{\text{CLMATSMP}(\min)}} \times 16 + 1 \end{aligned}$$

Minimum thresholds

The following restrictions must be taken into account:

- CLMAnCMPL \geq 0001_H
- CLMAnCMPH \geq CLMAnCMPL + 0003_H

31.5.2.2 Resetting CLMA_n

CLMA_n is initialized by the Power On Reset, the System Resets 1 and 2, the Application Reset 1, and register. CLMATEST.RESCLM can reset each CLMA_n by special procedure. See **Section 31.5.3.3, Resetting the Registers** and **Section 31.5.4.2, Reset Procedure Using CLMATEST.RESCLM** for details about CLMATEST.RESCLM.

Table 31.182 CLMA_n Reset Conditions

| CLMA | Reset Condition | | | | | | | |
|-------|-----------------|----------------|----------------|---------------------|-----------------|----------------|----------------|----------------|
| | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | CLMATEST.RESCLM | | | |
| | | | | | CLMA0TESEN = 1 | CLMA1TESEN = 1 | CLMA2TESEN = 1 | CLMA3TESEN = 1 |
| CLMA0 | √ | √ | √ | √ | √ | | | |
| CLMA1 | √ | √ | √ | √ | | √ | | |
| CLMA2 | √ | √ | √ | √ | | | √ | |
| CLMA3 | √ | √ | √ | √ | | | | √ |

31.5.2.3 Self-Diagnosis

Self-diagnosis of the clock monitor is available as described below.

- (1) Configure thresholds (CLMA_nCMPL/CLMA_nCMPH) of target CLMA. Set thresholds into values to generate an error.
- (2) Specify the clock monitor to be self-diagnosed.
Setting the CLMATEST.CLMA_nTESEN bit to 1 enables specifying the corresponding clock monitor to be self-diagnosed.
- (3) To prevent error notice to the ECM based on self-diagnosis, set CLMATEST.ERRMSK simultaneously with step (2).
- (4) To enable the monitor operation, Set CLMA_nCTL0.CLMA_nCLME bit to 1.
- (5) Wait for the time long enough to allow error occurrence, read the CLMATESTS register to see if an error has been generated in the clock monitor to be self-diagnosed. The time from the start of self-diagnosis to the error occurrence depends on the sampling period. A maximum of two sampling cycles are required.
- (6) Clear the error generated by self-diagnosis.
Setting CLMATEST.RESCLM to 1 enables initializing the clock monitor to be self-diagnosed.
- (7) Terminate self-diagnosis.
Setting all the bits in CLMATEST to 0 enables terminating self-diagnosis.

Before restarting the clock monitor that has been self-diagnosed, set the registers again as required.

31.5.3 Register

31.5.3.1 Register Protection

Write protected registers are protected from unintended write access due to factors such as program errors. The following clock monitor register features this special write protection function:

- CLMA_n control register 0 CLMA_nCTL0

Other registers can be protected from unintended write access due to factors such as program errors by configuration of the P-Bus Guards. For details, see **31.4.3, PBG**.

31.5.3.2 List of Registers

Table 31.183 Register Base Addresses

| Component | Symbol | Base Address |
|--------------------|--------------|------------------------|
| CLMA0 | <CLMA0_base> | FFF8 3100 _H |
| CLMA1 | <CLMA1_base> | FFF8 3200 _H |
| CLMA2 | <CLMA2_base> | FFF8 3300 _H |
| CLMA3 | <CLMA3_base> | FFF8 3400 _H |
| Clock Monitor Test | <CLMAT_base> | FFF8 3000 _H |

Each Clock Monitor is controlled and operated by the following registers.

Table 31.184 List of Registers

| Module Name | Address | Register Name | Description | Access Width | Value After Reset | Protection Register | |
|-------------------|---|------------------------|---|--------------|------------------------|------------------------|----------------------|
| | | | | | | Command Register | Status Register |
| CLMA _n | <CLMA _n _base> + 00 _H | CLMA _n CTL0 | CLMA _n control register 0 | 8 | 00 _H | CLMA _n PCMD | CLMA _n PS |
| CLMA _n | <CLMA _n _base> + 08 _H | CLMA _n CMPL | CLMA _n compare register L | 16 | 0001 _H | | |
| CLMA _n | <CLMA _n _base> + 0C _H | CLMA _n CMPH | CLMA _n compare register H | 16 | 03FF _H | | |
| CLMA _n | <CLMA _n _base> + 10 _H | CLMA _n PCMD | CLMA _n protection command register | 8 | 00 _H | | |
| CLMA _n | <CLMA _n _base> + 14 _H | CLMA _n PS | CLMA _n protection status register | 8 | 00 _H | | |
| CLMAC | <CLMAT_base> + 00 _H | CLMATEST | CLMA self-test register | 32 | 0000 0000 _H | | |
| CLMAC | <CLMAT_base> + 04 _H | CLMATESTS | CLMA self-test status register | 32 | 0000 0000 _H | | |

31.5.3.3 Resetting the Registers

Table 31.185 Register Reset Conditions

| Register Name | Reset Condition | | | |
|---------------|-----------------|----------------|----------------|---------------------|
| | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| All registers | √ | √ | √ | √ |

Each clock monitor (CLMA0/1/2/3) can also be reset by CLMATEST.RESCLM. In this case, it is necessary to perform a special procedure and reconfigure the CLMA parameters before the clock frequency is changed by the register for the operating clock.

For details about this special procedure, see **Section 31.5.4.2, Reset Procedure Using CLMATEST.RESCLM.**

Table 31.186 Individual Reset Conditions

| Register Name | CLMATEST.RESCLM Reset Target | | | |
|---------------|------------------------------|----------------|----------------|----------------|
| | CLMA0TESEN = 1 | CLMA1TESEN = 1 | CLMA2TESEN = 1 | CLMA3TESEN = 1 |
| CLMA0CTL0 | √ | | | |
| CLMA0CMPL | √ | | | |
| CLMA0CMPH | √ | | | |
| CLMA0PCMD | √ | | | |
| CLMA0PS | √ | | | |
| CLMA1CTL0 | | √ | | |
| CLMA1CMPL | | √ | | |
| CLMA1CMPH | | √ | | |
| CLMA1PCMD | | √ | | |
| CLMA1PS | | √ | | |
| CLMA2CTL0 | | | √ | |
| CLMA2CMPL | | | √ | |
| CLMA2CMPH | | | √ | |
| CLMA2PCMD | | | √ | |
| CLMA2PS | | | √ | |
| CLMA3CTL0 | | | | √ |
| CLMA3CMPL | | | | √ |
| CLMA3CMPH | | | | √ |
| CLMA3PCMD | | | | √ |
| CLMA3PS | | | | √ |
| CLMATEST | | | | |
| CLMATESTS | | | | |

31.5.3.4 CLMAnCTL0 — CLMAn Control Register 0

This register enables the clock monitor CLMAn.

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Protection

This register is write-protected; it can only be written by performing a special instruction sequence using the protection command register CLMAnPCMD. See **31.5.4.1, Write-Protected Registers** for details.

Access: These registers can be read/written in 8-bit units.

Address: <CLMAn_base> + 00_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------------|
| | — | — | — | — | — | — | — | CLMAn CLME |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 31.187 CLMAnCTL0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | CLMAnCLME | Enables or disables the clock monitor: 0: Disables CLMAn. 1: Enables CLMAn. This bit can only be cleared by a reset |

31.5.3.5 CLMAnCMPL — CLMAn Compare Register L

This register specifies the lower limit of monitored clock frequency.

Write access is permitted only when the CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: These registers can be read/written in 16-bit units.

Address: <CLMAn_base> + 08_H

Value after reset: 0001_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | CLMAnCMPL[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.188 CLMAnCMPL Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 to 0 | CLMAnCMPL [11:0] | Specifies the lower threshold The recommended value is $(f_{\text{CLMATMON}}(\text{min}) \times 16) / f_{\text{CLMATSMPL}}(\text{max}) - 1$ The minimum value is 0001 _H |

31.5.3.6 CLMAnCMPH — CLMAn Compare Register H

This register specifies the upper limit of monitored clock frequency.

Write access is permitted only when the CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: These registers can be read/written in 16-bit units.

Address: <CLMAn_base> + 0C_H

Value after reset: 03FF_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|----|----|----|----|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | — | — | — | — | CLMAnCMPH[11:0] | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.189 CLMAnCMPH Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 15 to 12 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 11 to 0 | CLMAnCMPH [11:0] | Specifies the upper threshold The recommended value is $(f_{\text{CLMATMON}}(\text{max}) \times 16) / f_{\text{CLMATSMPL}}(\text{min}) + 1$ The minimum value is CLMAnCMPL + 0003 _H |

31.5.3.7 CLMAnPCMD — CLMAn Protection Command Register

This register is a command register to protect the CLMAnCTL0 register from unintended write access. This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

NOTE

See **Section 31.5.4.1, Procedures to enable CLMAn** for the procedure.

Access: These registers can be written in 8-bit units.

Address: <CLMAn_base> + 10_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---------------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CLMAnREG[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |

Table 31.190 CLMAnPCMD Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 7 to 0 | CLMAnREG [7:0] | Protection command that enables writing to write-protected register (CLMAnCTL0). |

31.5.3.8 CLMAnPS — CLMAn Protection Status Register

This register is used to verify whether the write-protected register (CLMAnCTL0) has been successfully written or not.

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: These registers can be read in 8-bit units.

Address: <CLMAn_base> + 14_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|-------------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | CLMAn PRERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 31.191 CLMAnPS Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------|---|
| 7 to 1 | Reserved | When read, the value after reset is read. |
| 0 | CLMAnPRERR | Indicates whether the write-protected register (CLMAnCTL0) has been successfully written. 0: Write operation successful 1: Write operation failed |

31.5.3.9 CLMATEST — CLMA Self-test Register

This register is used for the self-test of the clock monitors. Each Clock Monitor can be tested individually.

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: These registers can be read/written in 32-bit units.

Address: <CLMAT_base> + 00_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----------------|----------------|----------------|----------------|------------|---------------|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | CLMA3 TESEN | CLMA2 TESEN | CLMA1 TESEN | CLMA0 TESEN | ERR MSK | MONCL KMSK | RES CLM |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 31.192 CLMATEST Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|--|
| 31 to 7 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 6 | CLMA3TESEN | CLMA3 self-test enable/disable 0: Test disabled. 1: Test enabled. |
| 5 | CLMA2TESEN | CLMA2 self-test enable/disable 0: Test disabled. 1: Test enabled. |
| 4 | CLMA1TESEN | CLMA1 self-test enable/disable 0: Test disabled. 1: Test enabled. |
| 3 | CLMA0TESEN | CLMA0 self-test enable/disable 0: Test disabled. 1: Test enabled. |
| 2 | ERRMSK ^{*1} | Masks an error notification to the ECM when CLMA _n detects an error. When the ERRMSK is set for a certain CLMA _n , the associated CLMA _n does not issue an error notification to the ECM even if it detects an error. 0: Does not mask an error notification to the ECM. 1: Masks an error notification to the ECM. |
| 1 | MONCLKMSK ^{*1} | Fixes the level of the clock input to the CLMA _n that should be monitored, to the low level. 0: Does not fix the clock input to the CLMA _n that should be monitored to the low level. 1: Fixes the clock input to the CLMA _n that should be monitored to the low level. |
| 0 | RESCLM ^{*1} | Initializes CLMA _n forcibly. 0: Does not initialize CLMA _n . 1: Initializes CLMA _n . |

Note 1. These bits are valid for CLMA_n which is in self-diagnosis mode by setting "1" to CLMA_nTESTEN.

NOTE

See Section 31.5.4.2, Reset Procedure Using CLMATEST.RESCLM for the individual reset procedure of each CLMA_n.

31.5.3.10 CLMATESTS — CLMA Self-test Status Register

This register is used for the self-test of the clock monitors. It monitors the error detection flags which are otherwise forwarded to the ECM module. Once error is detected, this register keeps the status until CLMA_n is reset.

This register is initialized by the Power On Reset, the System Resets 1 and 2, and the Application Reset 1.

Access: These registers can be read in 32-bit units.

Address: <CLMAT_base> + 04_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|------------|------------|------------|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | CLMA3 ERRS | CLMA2 ERRS | CLMA1 ERRS | CLMA0 ERRS |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.193 CLMATESTS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 4 | Reserved | When read, the value after reset is read. |
| 3 | CLMA3ERRS | Indicate whether or not CLMA _n have detected an error. These bits are not affected by CLMATEST.ERRMSK. 0: No error detected 1: Error detected |
| 2 | CLMA2ERRS | Indicate whether or not CLMA _n have detected an error. These bits are not affected by CLMATEST.ERRMSK. 0: No error detected 1: Error detected |
| 1 | CLMA1ERRS | Indicate whether or not CLMA _n have detected an error. These bits are not affected by CLMATEST.ERRMSK. 0: No error detected 1: Error detected |
| 0 | CLMA0ERRS | Indicate whether or not CLMA _n have detected an error. These bits are not affected by CLMATEST.ERRMSK. 0: No error detected 1: Error detected |

31.5.4 Operation

31.5.4.1 Write-Protected Registers

(1) Writing to Protected Registers

Writing to the CLMAnCTL0 register ($n = 0$ to 3) of each clock monitor is only possible with the protection unlock sequence described below.

Step 1. Write the fixed value (A5H) to the CLMAnPCMD register.

Step 2. Write the new setting to the CLMAnCTL0 register. Write the value after a reset to the reserved bits.

Step 3. Write the bitwise inverse of the setting value to the CLMAnCTL0 register. Write the inverse of the value after a reset to the reserved bits.

Step 4. Write the new setting to the CLMAnCTL0 register. Write the value after a reset to the reserved bits.

A value can only be written to a write-protected register by following the procedure above.

Protection is not unlocked if the procedure is not followed correctly, the setting is not written to the write-protected register, and the CLMAnPS.CLMAnPRERR bit is set to 1. (Although this is not a requirement, you can confirm if the values set in the targeted register was correctly written by checking that the setting of the CLMAnPS.CLMAnPRERR bit is 0 after step 4.)

In case of failure to follow the sequence, repeat the procedure from the beginning.

In case of writing to another register during steps 1 to 4 of the sequence, the protection function operates as follows. Operation is also as described when any interrupt is accepted during the sequence and the interrupt handling includes access to another register.

- When the writing is to another register in the same module, writing to the protected register fails and the CLMAnPS.CLMAnPRERR bit is set to 1.
- When writing is to a register in another module, writing to the protected register will successfully proceed to completion, if the remainder of the procedure is followed correctly

Reading of another register while the sequence is in progress does not lead to failure.

(2) Interrupt during Write Sequence

- (1) If an interrupt request is accepted during the write sequence and write access to a register of the same module is made:

The write to the write protection target register fails and the error monitor bit is set to 1. **Figure 31.10** shows an execution example.

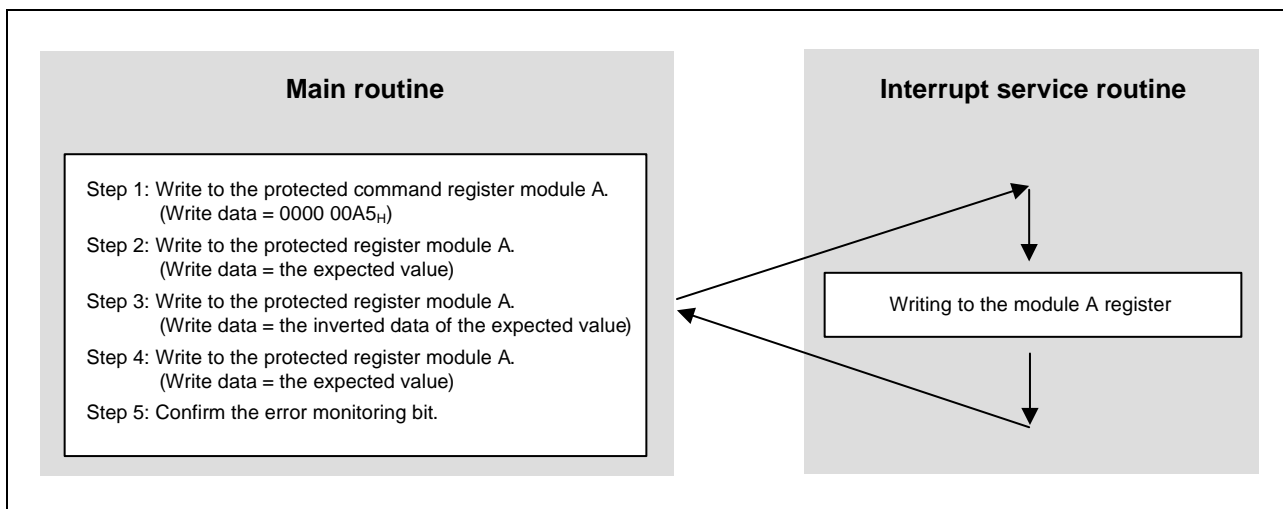


Figure 31.10 Example of Failed Write Sequence

- (2) If an interrupt request is accepted during the write sequence and write access to a register of a different module is made.

The write to the write protection register can be completed successfully. **Figure 31.11** shows an execution example.

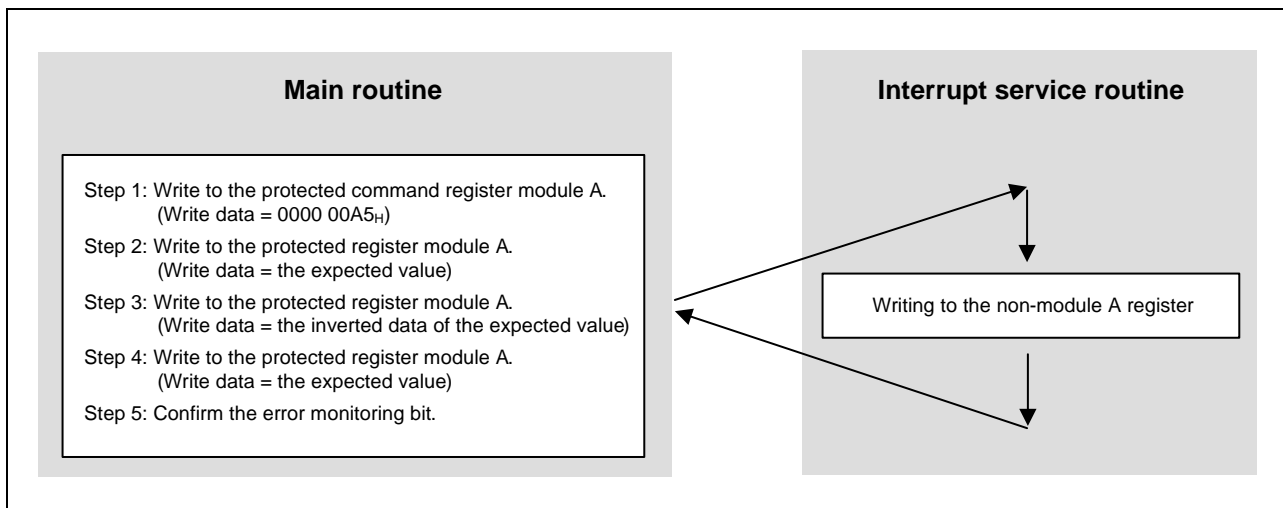


Figure 31.11 Example of Successful Write Sequence

(3) Emulation Break during Write Sequence

If an emulation break occurs during the above write sequence, e.g. because of a breakpoint hit, the register protection is suspended until normal operation is resumed. Even if any register of the same module is accessed during the break, the write sequence is not suspended and the error monitor bit is not set to 1.

31.5.4.2 Reset Procedure Using CLMATEST.RESCLM

Each clock monitor (CLMA0/1/2/3) can also be reset by CLMATEST.RESCLM. In this case, it is necessary to perform the following procedure and reconfigure the CLMA_n parameters before the clock frequency is changed by the register for the operating clock.

[Procedure]

Example: resetting CLMA0

1. CLMATEST.CLMA0TESEN = 1 (write data : 0000_0008_H)
2. CLMATEST.ERRMSK = 1 (write data : 0000_000C_H)
3. CLMATEST.MONCLKMSK = 1 (write data : 0000_000E_H)
4. CLMATEST.RESCLM = 1 (write data : 0000_000F_H)
5. CLMATEST.RESCLM = 0 (write data : 0000_000E_H)
6. CLMATEST.MONCLKMSK = 0 (write data : 0000_000C_H)
7. CLMATEST.ERRMSK = 0 (write data : 0000_0008_H)
8. CLMATEST.CLMA0TESEN = 0 (write data : 0000_0000_H)

31.6 BIST

This product incorporates the function to detect failures of the failure detection function itself, which is referred to as BIST. BIST execution results can be identified by the field BIST result register (BSEQ0ST).

31.6.1 List of Registers

Table 31.194 List of Registers

| Address | Symbol | Register Name | R/W | Value after reset | Access Size |
|------------------------|------------|---|-----|---------------------------------------|-------------|
| FFCD A000 _H | LBISTREF1 | Logic BIST reference value register 1 | R | 000A 5A5A _H * ¹ | 32 |
| FFCD A004 _H | LBISTREF2 | Logic BIST reference value register 2 | R | 0005 A5A5 _H * ¹ | 32 |
| FFCD A008 _H | MBISTREF | Memory BIST reference value register | R | 000A A55A _H * ¹ | 32 |
| FFCD A010 _H | LBISTSIG1 | Logic BIST signature value register 1 | R | 0005 A5A5 _H * ¹ | 32 |
| FFCD A014 _H | LBISTSIG2 | Logic BIST signature value register 2 | R | 000A 5A5A _H * ¹ | 32 |
| FFCD A018 _H | MBISTSIG | Memory BIST signature value register | R | 0005 5AA5 _H * ¹ | 32 |
| FFCD A020 _H | MBISTFTAGL | Memory BIST FTAG signature value register L | R | FFFF FFFF _H * ¹ | 32 |
| FFCD A024 _H | MBISTFTAGH | Memory BIST FTAG signature value register H | R | FFFF FFFF _H * ¹ | 32 |
| FFCD A030 _H | BSEQ0ST | BIST sequencer status register | R | 0000 0001 _H * ¹ | 32 |
| FFCD A034 _H | BSEQ0STB | BIST sequencer inverted status register | R | 0000 0002 _H * ¹ | 32 |
| FFCD A038 _H | BISTST | Field BIST Result register | R | 0000 0007 _H * ¹ | 32 |
| FFF8 0200 _H | BSEQ0CTL | Field BIST control register | R/W | 0000 0001 _H | 32 |

Note 1. This register doesn't show the fixed value.

31.6.2 Details of Registers

(1) LBISTREF1 — Logic BIST Reference Value Register 1

This register indicates the reference value of the Logic BIST.

Access: LBISTREF1 can be read only in 32-bit units.

Address: FFCD A000_H

Value after reset: *1

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|------------------|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | | | | LBISTREF1[19:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LBISTREF1[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.195 LBISTREF1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 20 | Reserved | When read, the value after reset is read. |
| 19 to 0 | LBISTREF1 [19:0] | Reference signature (expected signature) |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 It does not match the value of corresponding signature register.
 [Application Reset 1]
 The previous value is kept.

(2) LBISTREF2 — Logic BIST Reference Value Register 2

This register indicates the reference value of the Logic BIST.

Access: LBISTREF2 can be read only in 32-bit units.

Address: FFCD A004_H

Value after reset: *1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|------------------|----|----|----|
| | — | | | | | | | | | | | | LBISTREF2[19:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LBISTREF2[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.196 LBISTREF2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 20 | Reserved | When read, the value after reset is read. |
| 19 to 0 | LBISTREF2 [19:0] | Reference signature (expected signature) |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 It does not match the value of corresponding signature register.
 [Application Reset 1]
 The previous value is kept.

(3) MBISTREF — Memory BIST Reference Value Register

This register indicates the reference value of the Memory BIST.

Access: MBISTREF can be read only in 32-bit units.

Address: FFCD A008_H

Value after reset: *1

| | | | | | | | | | | | | | | | | |
|-------------------|----------------|----|----|----|----|----|----|----|----|----|----|----|-----------------|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | | | | | | | | | | | | MBISTREF[19:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MBISTREF[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.197 MBISTREF Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 31 to 20 | Reserved | When read, the value after reset is read. |
| 19 to 0 | MBISTREF [19:0] | Reference signature (expected signature) |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 It does not match the value of corresponding signature register.
 [Application Reset 1]
 The previous value is kept.

(4) LBISTSIG1 — Logic BIST Signature Value Register 1

This register indicates the signature value of the Logic BIST. Compare the reference signature of LBISTREF1 against the resulting signature of LBISTSIG1. The LBIST is passed if these are equal.

Access: LBISTSIG1 can be read only in 32-bit units.

Address: FFCD A010_H

Value after reset: *1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|------------------|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | LBISTSIG1[19:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LBISTSIG1[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.198 LBISTSIG1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 20 | Reserved | When read, the value after reset is read. |
| 19 to 0 | LBISTSIG1 [19:0] | LBIST1 signature result |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 It does not match the value of corresponding reference register.
 [Application Reset 1]
 The previous value is kept.

(5) LBISTSIG2 — Logic BIST Signature Value Register 2

This register indicates the signature value of the Logic BIST. Compare the reference signature of LBISTREF2 against the resulting signature of LBISTSIG2. The LBIST is passed if these are equal.

Access: LBISTSIG2 can be read only in 32-bit units.

Address: FFCD A014_H

Value after reset: *1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|------------------|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | LBISTSIG2[19:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | LBISTSIG2[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.199 LBISTSIG2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------|---|
| 31 to 20 | Reserved | When read, the value after reset is read. |
| 19 to 0 | LBISTSIG2 [19:0] | LBIST2 signature result |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 It does not match the value of corresponding reference register.
 [Application Reset 1]
 The previous value is kept.

(6) MBISTSIG — Memory BIST Signature Value Register

This register indicates the signature value of the Memory BIST. Compare the reference signature of MBISTREF against the resulting signature of MBISTSIG. The MBIST is passed if these are equal.

Access: MBISTSIG can be read only in 32-bit units.

Address: FFCD A018_H

Value after reset: *1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------------|----|----|----|----|----|----|----|----|----|----|----|-----------------|----|----|----|
| | — | — | — | — | — | — | — | — | — | — | — | — | MBISTSIG[19:16] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MBISTSIG[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.200 MBISTSIG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|---|
| 31 to 20 | Reserved | When read, the value after reset is read. |
| 19 to 0 | MBISTSIG [19:0] | MBIST1 signature result |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 It does not match the value of corresponding reference register.
 [Application Reset 1]
 The previous value is kept.

(7) MBISTFTAGL — Memory BIST FTAG Signature Value Register L

This register indicates the Memory BIST status of each RAM group (bridge).

Access: MBISTFTAGL can be read only in 32-bit units.

Address: FFCD A020_H

Value after reset: *1

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | MBISTFTAGL[31:16] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MBISTFTAGL[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.201 MBISTFTAGL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 31 to 0 | MBISTFTAGL [31:0] | The state of self-diagnostic MBIST of Bridge No.n 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to Table 31.202 for mapping of each RAM. |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 This register shows FFFFFFFF_H.
 [Application Reset 1]
 The previous value is kept.

Table 31.202 Mapping of MBISTFTAGL Register (1/2)

| Bit Position | Bit Name | Function |
|--------------|--------------------|--|
| 31 to 29 | MBISTFTAGL [31:29] | Reserved bit |
| 28 | MBISTFTAGL [28] | The state of self-diagnostic MBIST of Trace RAM0 0: MBIST status is PASS 1: MBIST status is FAIL |
| 27, 26 | MBISTFTAGL [27:26] | Reserved bit |
| 25 | MBISTFTAGL [25] | The state of self-diagnostic MBIST of CAN RAM (AFL). 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 24 | MBISTFTAGL [24] | The state of self-diagnostic MBIST of CAN RAM (MB). 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 23 to 21 | MBISTFTAGL [23:21] | Reserved bit |
| 20 | MBISTFTAGL [20] | The state of self-diagnostic MBIST of CSIH RAM. 0: MBIST status is PASS. 1: MBIST status is FAIL. |

Table 31.202 Mapping of MBISTFTAGL Register (2/2)

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 19 | MBISTFTAGL [19] | The state of self-diagnostic MBIST of FlexRay RAM (Temporary buffer RAM for FLX0). 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 18 | MBISTFTAGL [18] | The state of self-diagnostic MBIST of FlexRay RAM (Message RAM for FLX0). 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 17 | MBISTFTAGL [17] | Reserved bit |
| 16 | MBISTFTAGL [16] | The state of self-diagnostic MBIST of DTS RAM. 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 15 to 9 | MBISTFTAGL [15:9] | Reserved bit |
| 8 | MBISTFTAGL [8] | The state of self-diagnostic MBIST of Instruction cache RAM (data) (PE1). 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 7 | MBISTFTAGL [7] | The state of self-diagnostic MBIST of Instruction cache RAM (data) (PE1). 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 6 to 3 | MBISTFTAGL [6:3] | Reserved bit |
| 2 | MBISTFTAGL [2] | The state of self-diagnostic MBIST of Instruction cache RAM (tag) (PE1). 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 1, 0 | MBISTFTAGL [1:0] | Reserved bit |

(8) MBISTFTAGH — Memory BIST FTAG Signature Value Register H

This register indicates the Memory BIST status of each RAM group (bridge).

Access: MBISTFTAGH can be read only in 32-bit units.

Address: FFCD A024_H

Value after reset: *1

| | | | | | | | | | | | | | | | | |
|-------------------|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | MBISTFTAGH[31:16] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | MBISTFTAGH[15:0] | | | | | | | | | | | | | | | |
| Value after reset | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.203 MBISTFTAGH Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 31 to 0 | MBISTFTAGH [31:0] | The state of self-diagnostic MBIST of Bridge No.n 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to Table 31.204 for mapping of each RAM. |

Note 1. When the Field BIST is skipped, the value is shown as below.
[BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
[BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
This register shows FFFFFFFF_H.
[Application Reset 1]
The previous value is kept.

Table 31.204 Mapping of MBISTFTAGH Register (1/3)

| Bit Position | Bit Name | Function |
|--------------|-----------------|---|
| 31 | MBISTFTAGH [31] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBF_0000 _H to FEBF_FFFC _H and the 4 bits of address LSB are 8 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 30 | MBISTFTAGH [30] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBF_0000 _H to FEBF_FFFC _H and the 4 bits of address LSB are 0 _H or 4 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 29 | MBISTFTAGH [29] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBF_0000 _H to FEBF_FFFC _H and the 4 bits of address LSB are 8 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 28 | MBISTFTAGH [28] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBF_0000 _H to FEBF_FFFC _H and the 4 bits of address LSB are 0 _H or 4 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |

Table 31.204 Mapping of MBISTFTAGH Register (2/3)

| Bit Position | Bit Name | Function |
|--------------|-----------------------|---|
| 27 | MBISTFTAGH [27] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBE_0000 _H to FEBE_FFFC _H and the 4 bits of address LSB are 8 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 26 | MBISTFTAGH [26] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBE_0000 _H to FEBE_FFFC _H and the 4 bits of address LSB are 0 _H or 4 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 25 | MBISTFTAGH [25] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBE_0000 _H to FEBE_FFFC _H and the 4 bits of address LSB are 8 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 24 | MBISTFTAGH [24] | The state of self-diagnostic MBIST of Local RAM(PE1) for which the address is in the range of FEBE_0000 _H to FEBE_FFFC _H and the 4 bits of address LSB are 0 _H or 4 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 23 to 20 | MBISTFTAGH [23:20] | Reserved bit |
| 19 | MBISTFTAGH [19] | The state of self-diagnostic MBIST of ERAM for which the address is in the range of F982_0000 _H to F982_1FFC _H and the 4 bits of address LSB are 8 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 18 | MBISTFTAGH [18] | The state of self-diagnostic MBIST of ERAM for which the address is in the range of F982_0000 _H to F982_1FFC _H and the 4 bits of address LSB are 0 _H or 4 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 17 | MBISTFTAGH [17] | The state of self-diagnostic MBIST of ERAM for which the address is in the range of F980_0000 _H to F980_1FFC _H and the 4 bits of address LSB are 8 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 16 | MBISTFTAGH [16] | The state of self-diagnostic MBIST of ERAM for which the address is in the range of F980_0000 _H to F980_1FFC _H and the 4 bits of address LSB are 0 _H or 4 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 15 to 10 | MBISTFTAGH [15:10] | Reserved bit |
| 9 | MBISTFTAGH [9] | The state of self-diagnostic MBIST of Global RAM for which the address is in the range of FEF0_0000 _H to FEF0_7FFC _H and the 4 bits of address LSB are 4 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 8 | MBISTFTAGH [8] | The state of self-diagnostic MBIST of Global RAM for which the address is in the range of FEF0_0000 _H to FEF0_7FFC _H and the 4 bits of address LSB are 0 _H or 8 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |
| 7 | MBISTFTAGH [7] | The state of self-diagnostic MBIST of Global RAM for which the address is in the range of FEEF_8000 _H to FEEF_FFFC _H and the 4 bits of address LSB are 4 _H or C _H . 0: MBIST status is PASS. 1: MBIST status is FAIL. |

Table 31.204 Mapping of MBISTFTAGH Register (3/3)

| Bit Position | Bit Name | Function |
|--------------|---------------------|--|
| 6 | MBISTFTAGH [6] | The state of self-diagnostic MBIST of Global RAM for which the address is in the range of FEEF_8000 _H to FEF0_FFFC _H and the 4 bits of address LSB are 0 _H or 8 _H . 0: MBIST status is PASS. 1: MBIST status is FAIL |
| 5 to 0 | MBISTFTAGH [5:0] | Reserved bit |

(9) BSEQ0ST — BIST Sequencer Status Register

This register indicates the state of the BIST sequencer.

Access: BSEQ0ST can be read only in 32-bit units.

Address: FFCD A030_H

Value after reset: *1

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | BIST_RESULT [1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.205 BSEQ0ST Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|--|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1, 0 | BIST_RESULT [1:0] | 10: The self-diagnostic BIST is normally ended. Other than above: The self-diagnostic BIST is abnormally ended, not completed, nor not started. |

- Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2]
 This register shows 00000001_H.
 [Debug mode (DCUTRST = 1)]
 This register shows 00000002_H.
 [Application Reset 1]
 The previous value is kept.

(10) BSEQ0STB— BIST Sequencer Inverted Status Register

This register indicates the inverted state of the BIST sequencer

Access: BSEQ0STB can be read only in 32-bit units.

Address: FFCD A034_H

Value after reset: *1

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | BIST_RESULTB [1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.206 BSEQ0STB Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------|---|
| 31 to 2 | Reserved | When read, the value after reset is read. |
| 1, 0 | BIST_RESULTB [1:0] | Indicates the inverse of the BIST_RESULT[1:0] bits in the BSEQ0ST register. |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2]
 This register shows 00000002_H.
 [Debug mode ($\overline{\text{DCUTRST}} = 1$)]
 This register shows 00000001_H.
 [Application Reset 1]
 The previous value is kept.

(11) BISTST— Field BIST Result Register

This register indicates the result of the field BIST.

Access: BISTST can be read only in 32-bit units.

Address: FFCD A038_H

Value after reset: *1

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|--------------|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | MBIST ST | LBIST2 ST | LBIST1 ST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | *1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 31.207 BISTST Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 3 | Reserved | When read, the value after reset is read. |
| 2 | MBISTST | 0: MBIST passed successfully 1: MBIST has detected an error |
| 1 | LBIST2ST | 0: LBIST2 passed successfully 1: LBIST2 has detected an error |
| 0 | LBIST1ST | 0: LBIST1 passed successfully 1: LBIST1 has detected an error |

Note 1. When the Field BIST is skipped, the value is shown as below.
 [BSEQ0CTL.HWBISTEXE = 0 and System reset1 (Pin reset only)] or
 [BSEQ0CTL.HWBISTEXE = 0 and System Reset 2] or [Debug mode (DCUTRST = 1)]
 This register shows 00000007_H.
 [Application Reset 1]
 The previous value is kept.

(12) BSEQ0CTL — Field BIST Control Register

This register is used to control the field BIST.

Access: BSEQ0CTL can be read/written in 32-bit units.

Address: FFF8 0200_H

Value after reset: 0000 0001_H This register is reset by a Power On Reset or a System Reset 1 (CVM reset only)

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | HWBIS TEXE |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 31.208 BSEQ0CTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | HWBISTEXE | This bit controls whether to execute the Field BIST or not after System Reset 1 (except CVM reset) and System Reset 2. 0: Skip BIST 1: Execute BIST |

31.6.3 Confirming Completion of BIST with no Abnormalities Detected

The registers listed below can be used to confirm the result of field BIST.

- Confirming matching of field BIST signatures

The expected value stored in logic BIST signature register 1 or 2 or in the memory BIST signature register matching the value of the result stored in the logic BIST signature result register 1 or 2 or in the memory BIST signature result register indicates that no abnormalities were detected by field BIST.

LBISTREF1 = LBISTSIG1

LBISTREF2 = LBISTSIG2

MBISTREF = MBISTSIG

- Confirming the error status

Use the BIST error status register to confirm that the BIST sequence has been completed without detecting abnormalities. The BSEQ0STB register value indicates the inverse of the BSEQ0ST register value.

BSEQ0ST = 0000 0002_H

- Debug mode:

In debug mode ($\overline{\text{DCUTRST}} = \text{H}$), expected values are not stored in the LBISTREF1, LBISTREF2 and MBISTREF registers, and results are not stored in the LBISTSIG1, LBISTSIG2, and MBISTSIG registers. Design the program to judge the result of field BIST from the value of $\overline{\text{DCUTRST}}$ (JP0_4) pin.

31.6.4 State Transition of Field BIST

The following chart shows the state transition of the Field BIST.

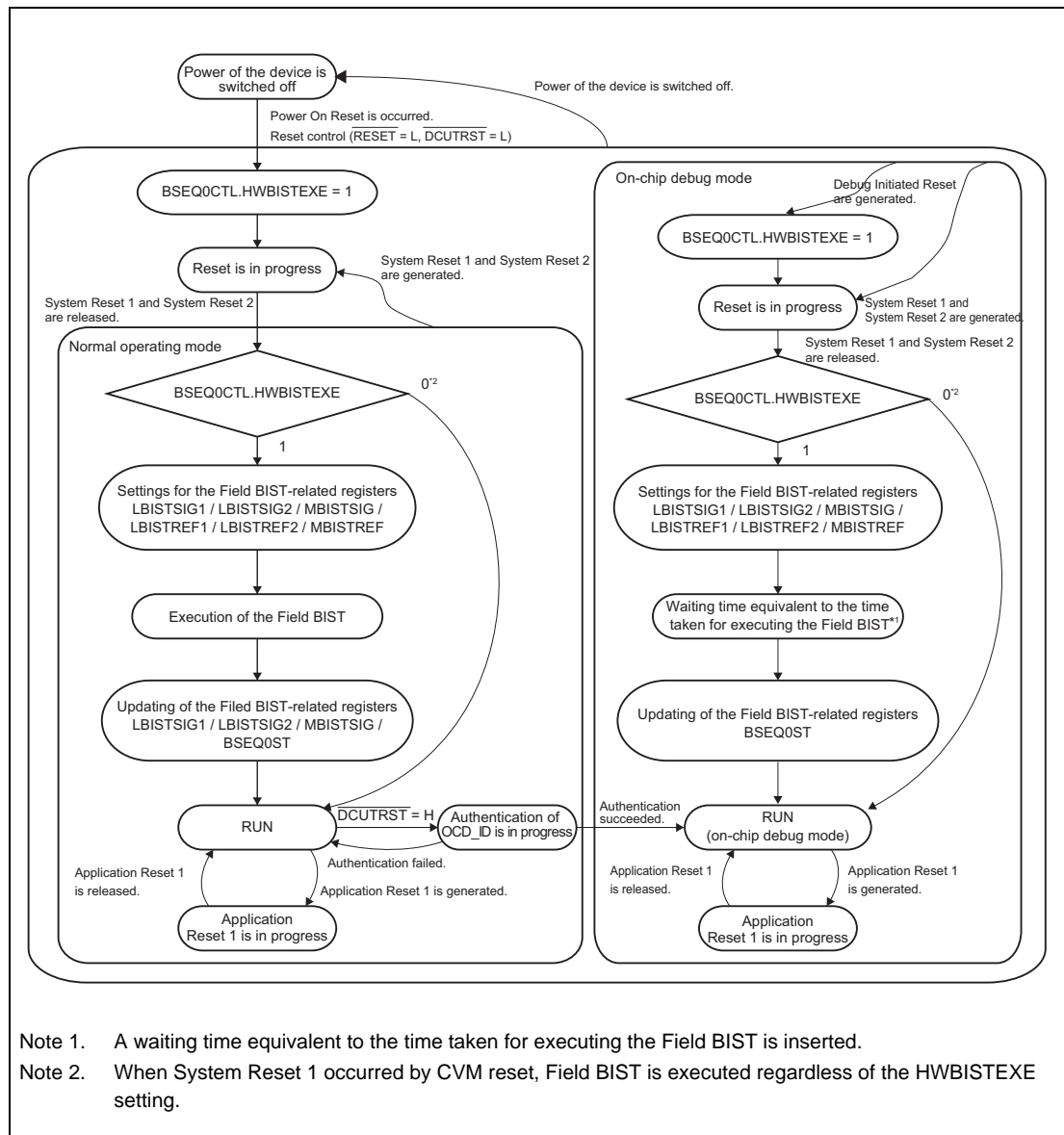


Figure 31.12 State Transition of Field BIST

31.6.5 Note

In this device, when executing the Field BIST, test code is read at the beginning of the program. If ECC 1-bit error or ECC 2-bit error is detected during reading of the test code, the error is notified to ECM. For details, see **Table 32.9, List of Error Inputs**. Field BIST is executed at the time of ECC error occurrence, but MCU does not move to normal operation. When ECC 1-bit error detected, the MCU does not correct ECC error.

31.7 ECM

The ECM monitors various failure detection states in the LSI chip, and defines the operation to be carried out upon failure detection. For the details of the ECM, refer to **Section 32, Error Control Module (ECM)**.

31.8 CVM

The core voltage monitor (CVM) detects over and under voltage of the core voltage. For the details of the CVM, refer to **Section 10, Core Voltage Monitor**.

31.9 WDTA

The window watchdog timer (WDTA) detects deadlock of CPU operation. For the details of the WDTA, refer to **Section 21, Window Watchdog Timer A (WDTA)**.

31.10 DCRA

The Data CRC Function A can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths. For the details of the DCRA, refer to **Section 33, Data CRC Function A (DCRA)**.

Section 32 Error Control Module (ECM)

32.1 Features of RH850/P1M-E ECM

32.1.1 Units

This microcontroller has the following number of ECM units.

Each ECM unit has one channel interface.

Table 32.1 Unit Configurations and Channels

| Unit Name (Channel Name) ECM | Channels per Unit |
|------------------------------|-------------------|
| ECM | 1 |

Table 32.2 Index

| Index | Meaning |
|-------|---|
| m | Throughout this section, the individual ECM Master and ECM Checker are identified by the index "m" (m = M, C): for example, ECMmEST bit is the ECM m Error set trigger bit. |

32.1.2 Register Base Address

ECM base addresses are listed in the following table.

ECM register addresses are given as offsets from the base addresses in general.

Table 32.3 Register Base Address

| Base Address | Name Base Address |
|--------------|------------------------|
| <ECMM_base> | FFD6 0000 _H |
| <ECMC_base> | FFD6 1000 _H |
| <ECM_base> | FFD6 2000 _H |

32.1.3 Clock Supply

Clock supply by and to ECM is listed in the following table.

Table 32.4 Clock Supply

| Unit Name | Clock port | Explanation | Specification | |
|-----------|------------|--|-----------------------------|-----------|
| | | | Clock name | Symbol |
| ECM | PCLK | Peripheral Bus clock. | High speed peripheral clock | CLK_HSB |
| | cntclk | Delay timer/Error clear mask timer clock 4 MHz, 1/2 the frequency of CLK_IOSC. | Delay Timer Clock | CLK_IOSC2 |

32.1.4 Interrupt Requests

ECM interrupt requests are listed in the following table. The interrupt request signal is driven to the high level with a pulse width of one PCLK cycle when error source status whose interrupt generation is enabled is set.

Table 32.5 Interrupt Requests

| Unit Interrupt Name | Description | Interrupt Number |
|---------------------|-----------------------------------|------------------|
| INTECM | ECM maskable interrupt (EI level) | 8 |
| INTECMNMI | ECM non-maskable interrupt | FENMI |

32.1.5 External Output Signals

External output signals of ECM are listed below.

Table 32.6 External Output Signals

| Unit Signal Name | Description | Alternative port pin signal |
|---------------------------------|--|---------------------------------|
| $\overline{\text{ERROROUT}}$ | <p>ERROROUT mode can be selected out of two modes:</p> <ol style="list-style-type: none"> 1. ERROROUT from ECM master is singly output to the ERROROUT pin 2. Logical AND of ERROROUT from ECM master and ERROROUT_C from ECM checker is output to the ERROROUT pin. | $\overline{\text{ERROROUT}}$ |
| $\overline{\text{ERROROUT_C}}$ | Error output signal from ECM checker | $\overline{\text{ERROROUT_C}}$ |

32.2 Overview

32.2.1 Specification Overview

ECM (Error Control Module) collects error signals coming from various error sources and monitoring circuits. It also outputs error signals from the error pins ($\overline{\text{ERROROUT}}$ and $\overline{\text{ERROROUT_C}}$) and generates interrupts and Error Control Module Reset signals. **Table 32.7** shows the specification overview of ECM.

Table 32.7 Specification Overview

| Item | Description |
|--|---|
| Safety processing | <p>ECM can handle the following processing in response to error signal inputs from individual modules.</p> <ul style="list-style-type: none"> • Error flag set • EI level interrupt generation EI level interrupt generation can be controlled (enabled/disabled) for individual errors. • Non-maskable interrupt generation Non-maskable interrupt generation can be controlled (enabled/disabled) for individual errors. • Internal reset generation Error Control Module Reset signals can be controlled (enabled/disabled) for individual errors. • Error pin outputs Pin output masks can be controlled (enabled/disabled) for individual errors. Outputs can be toggled in response to a timer input or made at a fixed level. |
| Error status | <p>ECM incorporates error source status registers, which can be used to confirm the error status from the error flag.</p> <p>The error flags are only cleared by software or a power on reset. In case of resets other than power on reset, the error flags are kept and the reset generation source can be confirmed by reading the status register after reset.</p> |
| Debug, self-diagnosis | <ul style="list-style-type: none"> • Pseudo errors can be generated for debug and self-diagnosis. The operation in response to the injection of pseudo errors is identical to that for the occurrence of real errors. All configurations for the masking of error pin output, interrupt, or Error Control Module Reset apply in the same way. • ECM incorporates a loop-back function of the error pin outputs that are used to diagnose the paths to the error output pins. The status of the error output pins is reflected to internal registers and can be confirmed by reading the registers. |
| Timeout function | <p>ECM incorporates a function that generates error signals output(s) or Error Control Module Reset when the count value of the delay timer matches with the delay timer compare register because the delay timer was not stopped during the interrupt processing after being started simultaneously with the occurrence of an interrupt request.</p> <p>If another error by which the delay timer is to be started occurs while the delay timer is running, the current value counted by the delay timer is not reset but the timer continues to run. An ECM reset is generated if the counter overflows before counting by the delay timer is stopped.</p> |
| Register protection | <p>A write-protection with Protection Unlock Sequence is incorporated to protect registers from illegal write access.</p> |
| $\overline{\text{ERROROUT}}$ and $\overline{\text{ERROROUT_C}}$ clear masking | <p>ECM incorporates a function that can mask software clearance of $\overline{\text{ERROROUT}}$ and $\overline{\text{ERROROUT_C}}$ until the counter which is started from error occurrence reaches the value specified in the Error Output Clear Invalidation Configuration register. If another error occurs during time counting, then the time count is reset and restarted from 0.</p> |
| Others | <p>ECM is duplexed. ECM incorporates the error output pins.</p> <p>The $\overline{\text{ERROROUT}}$ outputs from the ECM master and ECM checker are constantly compared. If they do not match, an ECM compare error (error source 92) occurs.</p> |

32.2.2 Flash Option

Table 32.8 Flash Option

| Flash Option | Function | Description | Option Byte |
|--------------|-----------------------|---|----------------|
| ERROUTSEL | ERROROUT Mode setting | Specifies the ERROROUT mode settings: 0: ERROROUT from ECM master is singly output to ERROROUT pin. 1: Logical AND of ERROROUT from ECM master and ERROROUT_C from ECM checker is output to ERROROUT pin. | OPBT0.OPBT0[1] |

32.3 Block Diagram

ECM is redundantly implemented using ECM Master and ECM Checker. The ERROROUT signal to PIC1 is not toggled, even if the ECM is set for dynamic mode. See **Figure 32.1, Connection between ECM Master, ECM Checker and Peripherals.**

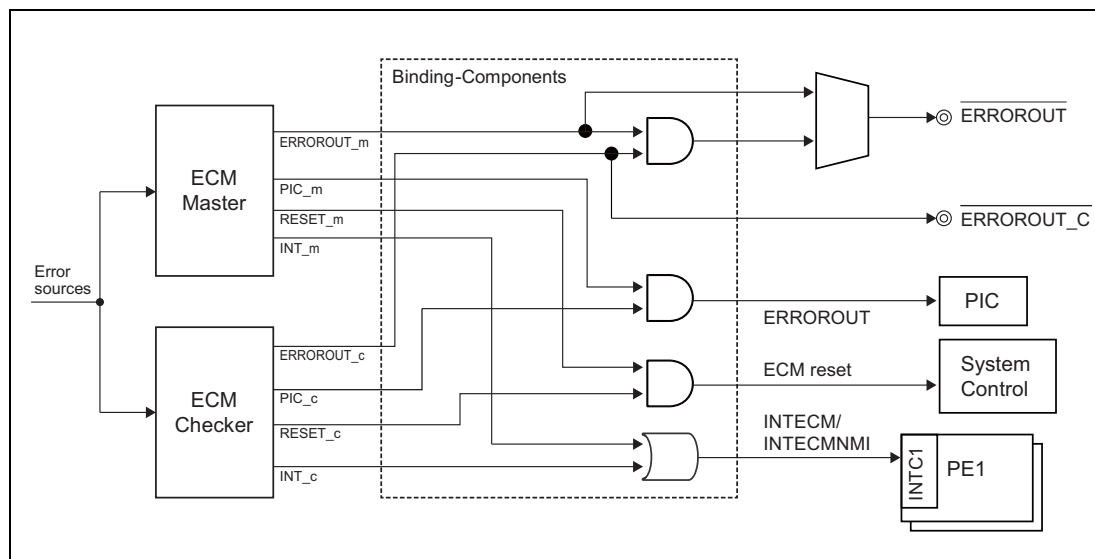


Figure 32.1 Connection between ECM Master, ECM Checker and Peripherals

32.3.1 Error Input

Table 32.9 shows the error inputs to ECM of RH850/P1M-E.

Table 32.9 List of Error Inputs (1/4)

| No. | Error sources | Module | Comment |
|-----|---|---|--|
| 0 | Window watchdog timer error | WDTA | Occurs when WDTA detects an error. (For details, see Section 21.5.3, WDTA Error Detection.) |
| 1 | DCLS compare error | CPU | Occurs when outputs of the CPU master core and checker core mismatch (Lock Step). |
| 2 | PFSS compare error <ul style="list-style-type: none"> DMA GRAM Controller | DMA/GRAM controller | <ul style="list-style-type: none"> Occurs when outputs of the redundant DMA mismatch (Lock Step). Occurs when outputs of the redundant function blocks in GRAM controller mismatch. |
| 3 | Bus bridge error | System Interconnect / Code Flash Interface / GRAM Interface / P-Bus | Occurs when abnormal arbitration detected in bus arbiter. |
| 4 | Compare error of functional block with redundancy | System Interconnect / Code Flash Interface / GRAM Interface / P-Bus | <ul style="list-style-type: none"> Occurs when the ECC decoder for internal bus detected ECC error output mismatch. Occurs when the address parity checker detected parity error output mismatch. Occurs when the comparator for lock step function detected compare error output mismatch. |
| 5 | Reserved | — | — |
| 6 | Temperature sensor error (over/under temp. detection) | TSN0 | Occurs when the temperature sensor detects an abnormal temperature error. (For details, see Section 11, Temperature Sensor.) |
| 7 | Reserved | — | — |

Table 32.9 List of Error Inputs (2/4)

| No. | Error sources | Module | Comment |
|-----|--|-------------------------------|--|
| 8 | Clock monitor upper-limit error for Main OSC | CLMA | Occurs when the clock monitor detects abnormal frequency of the targeted clock. (For details, see Section 31.5, Clock Monitor (CLMA).) |
| 9 | Clock monitor lower-limit error for Main OSC | | |
| 10 | Clock monitor upper-limit error for WDTA count clock | | |
| 11 | Clock monitor lower-limit error for WDTA count clock | | |
| 12 | Clock monitor upper-limit error for Peripheral clock | | |
| 13 | Clock monitor lower-limit error for peripheral clock | | |
| 14 | Clock monitor upper-limit error for PE1 checker core clock | | |
| 15 | Clock monitor lower-limit error for PE1 checker core clock | | |
| 16 | Local RAM ECC <ul style="list-style-type: none"> • uncorrectable error | Local RAM | Occurs when local RAM ECC decoder detects a data ECC 2-bit error. |
| 17 | Global RAM ECC <ul style="list-style-type: none"> • uncorrectable error | Global RAM | Occurs when global RAM ECC decoder detects a data ECC 2-bit error or an address ECC 2-bit error. |
| 18 | Instruction Cache RAM EDC <ul style="list-style-type: none"> • uncorrectable error | Instruction Cache RAM | Occurs when instruction cache RAM EDC decoder detects a data 1-bit EDC error or a data EDC 2-bit error. |
| 19 | Code Flash ECC/address parity error <ul style="list-style-type: none"> • uncorrectable error and address parity error | Code Flash | Occurs when code flash ECC decoder detects a data ECC 2-bit or address parity error. |
| 20 | Data Flash ECC <ul style="list-style-type: none"> • uncorrectable error | Data Flash | Occurs when data flash ECC decoder detects a data ECC 2-bit error. |
| 21 | Peripheral (CSIH:SPI) RAM ECC <ul style="list-style-type: none"> • uncorrectable error | CSIH | Occurs when data is read from CSIH RAM and if a data ECC 2-bit error is detected. |
| 22 | Peripheral (RS-CANFD) RAM ECC <ul style="list-style-type: none"> • uncorrectable error | RS-CANFD | Occurs when data is read from RS-CANFD RAM and if a data ECC 2-bit error is detected. |
| 23 | Reserved | — | — |
| 24 | Peripheral (FlexRay) RAM ECC <ul style="list-style-type: none"> • uncorrectable error | FLXA | Occurs when data is read from FlexRay RAM and if a data ECC 2-bit error is detected. |
| 25 | Reserved | — | — |
| 26 | Reserved | — | — |
| 27 | Reserved | — | — |
| 28 | Bus ECC error: DED | System Interconnect/ P-Bus | Occurs when a data ECC 2-bit error or an address ECC 2-bit error is detected in ECC decoder for the data transfer path. |
| 29 | Bus ECC error: SED | System Interconnect/ P-Bus | Occurs when a data ECC 1-bit error or an address ECC 1-bit error is detected in ECC decoder for the data transfer path. |
| 30 | Reserved | — | — |
| 31 | Reserved | — | — |
| 32 | Local RAM error address overflow | Local RAM | Occurs when a address overflow error for the local RAM data ECC 1-bit error is detected. |
| 33 | Global RAM error address overflow | Global RAM | Occurs when a address overflow error for the global RAM data ECC 1-bit error is detected. |
| 34 | Reserved | — | — |
| 35 | Code Flash ECC error address overflow | Code Flash | Occurs when a address overflow error for the code flash data ECC 1-bit error is detected. |

Table 32.9 List of Error Inputs (3/4)

| No. | Error sources | Module | Comment |
|-----|--|--------------------|--|
| 36 | Data Flash ECC error address overflow | Data Flash | Occurs when a address overflow error for the code flash data ECC error is detected. |
| 37 | Peripheral RAM ECC error address overflow | CSIH/RS-CANFD/FLXA | Occurs when a address overflow error for the CSIH/RS-CANFD/FLXA RAM data ECC error is detected. |
| 38 | Reserved | — | — |
| 39 | Reserved | — | — |
| 40 | DTSRAM ECC <ul style="list-style-type: none"> • uncorrectable error | DTS | Occurs when data is read from DTS RAM and if a data ECC 2-bit error is detected. |
| 41 | DTSRAM ECC <ul style="list-style-type: none"> • correctable error*¹ | DTS | Occurs when data is read from DTS RAM and if a data ECC 1-bit error is detected. |
| 42 | Reserved | — | — |
| 43 | Reserved | — | — |
| 44 | Reserved | — | — |
| 45 | Reserved | — | — |
| 46 | Reserved | — | — |
| 47 | Reserved | — | — |
| 48 | Local RAM ECC <ul style="list-style-type: none"> • correctable error*¹ | Local RAM | Occurs when local RAM ECC decoder detects a data ECC 1-bit error. |
| 49 | Global RAM ECC <ul style="list-style-type: none"> • correctable error*¹ | Global RAM | Occurs when global RAM ECC decoder detects a data ECC 1-bit error or an address ECC 1-bit error. |
| 50 | Reserved | — | — |
| 51 | Code Flash ECC <ul style="list-style-type: none"> • correctable error*¹ | Code Flash | Occurs when code flash ECC decoder detects a data ECC 1-bit error. |
| 52 | Data Flash ECC <ul style="list-style-type: none"> • correctable error*¹ | Data Flash | Occurs when data flash ECC decoder detects a data ECC 1-bit error. |
| 53 | Peripheral (CSIH:SPI) RAM ECC <ul style="list-style-type: none"> • correctable error*¹ | CSIH | Occurs when data is read from CSIH RAM and if a data ECC 1-bit error is detected. |
| 54 | Peripheral (RS-CANFD) RAM ECC <ul style="list-style-type: none"> • correctable error*¹ | RS-CANFD | Occurs when data is read from RS-CANFD RAM and if a data ECC 1-bit error is detected. |
| 55 | Reserved | — | — |
| 56 | Peripheral (FlexRay) RAM ECC <ul style="list-style-type: none"> • correctable error*¹ | FLXA | Occurs when data is read from FlexRay RAM and if a data ECC 1-bit error is detected. |
| 57 | Reserved | — | — |
| 58 | Reserved | — | — |
| 59 | Reserved | — | — |
| 60 | Reserved | — | — |
| 61 | Reserved | — | — |
| 62 | Reserved | — | — |
| 63 | Reserved | — | — |
| 64 | PE guard error | CPU | Occurs when PE guard function (PEG) detects a PE guard violation. (For details, See Section 3.2.3.1, PE Guard Function (PEG).) |
| 65 | Global RAM guard error | Global RAM | Occurs when GRAM guard function (GRG) detects a GRAM guard violation. (For details, see Section 31.4.1.3, Global RAM Protection and Section 31.4.2, GRG.) |
| 66 | Reserved | — | — |

Table 32.9 List of Error Inputs (4/4)

| No. | Error sources | Module | Comment |
|-----|---|----------------|---|
| 67 | Slave guard error (P-Bus Guard + H-Bus Guard) | P-Bus / H-Bus | Occurs when P-Bus guard (PBG) or H-Bus guard (HBG) function detects a bus guard violation. (For details, see Section 31.4.1.2, Peripheral Register Protection with H-bus Guard (HBG) and P-bus Guard (PBG) , Section 31.4.3, PBG and Section 31.4.4, HBG .) |
| 68 | Reserved | — | — |
| 69 | Reserved | — | — |
| 70 | Reserved | — | — |
| 71 | Reserved | — | — |
| 72 | Reserved | — | — |
| 73 | Reserved | — | — |
| 74 | Reserved | — | — |
| 75 | Reserved | — | — |
| 76 | Reserved | — | — |
| 77 | Reserved | — | — |
| 78 | Reserved | — | — |
| 79 | Reserved | — | — |
| 80 | DMA transfer error | DMA | Occurs when a DMA transfer error is detected. (For details, see Section 7.5.2, DMA Transfer Error .) |
| 81 | DMA illegal access error | DMA | Occurs when register access protection function in DMA detects an illegal access. (For details, see Section 7.6, Reliability Function .) |
| 82 | Flash <ul style="list-style-type: none"> Flash sequencer error | Flash | Occurs when code flash or data flash illegal access detected. (For details, <i>Flash Memory Users Manual Section 4.2 Flash Access Status Register (FASTAT)</i> , <i>4.3 Flash Access Error Interrupt Enable Register (FAEINT)</i> .) |
| 83 | Flash <ul style="list-style-type: none"> ECC correctable/uncorrectable errors during FACI reset ECC correctable/uncorrectable errors of the BIST code | Flash | Occurs when a FACI reset transfer error or ECC error of BIST code is detected. (About a FACI reset transfer error, <i>Flash Memory Users Manual Section 4.12 FACI Reset Transfer Status Register (FRTSTAT)</i> , <i>4.13 FACI Reset Transfer Error Interrupt Enable Register (FRTEINT)</i> .) |
| 84 | A/D Converter parity error | ADCG | Occurs when ADCA detects a parity error. (For details, Section 30.4.5.2, A/D Error Interrupt Request and A/D Parity Error Interrupt Request .) |
| 85 | Reserved | — | — |
| 86 | Reserved | — | — |
| 87 | Reserved | — | — |
| 88 | Mode error <ul style="list-style-type: none"> Unintended deactivation of user mode | Operating Mode | Occurs at an unintentional disabling of normal operating mode while normal operating mode is set (FLMD0 = 0). |
| 89 | Mode error <ul style="list-style-type: none"> Unintended activation of Code Flash Programming mode | Operating Mode | Occurs at an unintentional transition to serial programming mode while normal operating mode is set (FLMD0 = 0). |
| 90 | Mode error <ul style="list-style-type: none"> Unintended Debug Enable detection | Operating Mode | Occurs at an unintentional transition to debug mode while normal operating mode is set (FLMD0 = 0). |
| 91 | FieldBIST executed*2 or following mode error <ul style="list-style-type: none"> Unintended activation of Test Mode | Operating Mode | <ul style="list-style-type: none"> Occurs at an unintentional transition to production test mode while normal operating mode is set (FLMD0 = 0). Occurs when Field BIST operation is ended. |
| 92 | ECM compare error | ECM | Occurs when ERROROUT output of the ECM master and ECM checker mismatched. |

Note 1. Even if correction is disabled, single bit errors are still reported as correctable errors to the ECM.

Note 2. Even when Field BIST operation has normally ended, an error is notified error to ECM.

32.3.2 Operations for Error Output

After reset release, the $\overline{\text{ERROROUT}}$ pin and $\overline{\text{ERROROUT_C}}$ pin output the low (error) level. Follow the procedure described in **Section 32.4.3, ECMmECLR — ECM Master/Checker Error Clear Trigger Register (m = M/C)**, to clear the error before using ECM.

The error output can be configured for two different modes of operation, non-dynamic or dynamic.

Table 32.10 $\overline{\text{ERROROUT}}$ Output Operation

| Error Status ECMmSSE031 to ECMmSSE000 ECMmSSE131 to ECMmSSE100 ECMmSSE230 to ECMmSSE200 | Operating Mode ECMSL0 Bit | Error Output Operating Mode | Error Output Level | Error Status |
|--|------------------------------|--------------------------------|---------------------------------------|--------------|
| | | | | |
| 0 | 0 | Non-dynamic | H | No error |
| | 1 | Dynamic | Toggles (according to timer input) | No error |
| 1 | 0 | Non-dynamic | L | Error |
| | 1 | Dynamic | L | Error |

32.3.3 $\overline{\text{ERROROUT}}$ behavior at reset

The following table explains the behavior of the error output logic, the $\overline{\text{ERROROUT}}$ pin and the $\overline{\text{ERROROUT_C}}$ pin at reset. Also the level of the $\overline{\text{ERROROUT}}$ signal and the $\overline{\text{ERROROUT_C}}$ signal during and after reset are explained.

Table 32.11 $\overline{\text{ERROROUT}}$ and $\overline{\text{ERROROUT_C}}$ behavior at reset

| Category | Reset signal for initialize | | | |
|--|-----------------------------|--------------------|----------------|--|
| | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 |
| $\overline{\text{ERROROUT}}$ pin level during reset | Hi-Z | Hi-Z ^{*1} | Low level | No effect |
| $\overline{\text{ERROROUT}}$ pin level after reset | Low level | Low level | Low level | Level according to error status before reset |
| $\overline{\text{ERROROUT_C}}$ pin level during reset | Hi-Z (GPIO) | Hi-Z (GPIO) | Hi-Z (GPIO) | Hi-Z (GPIO) |
| $\overline{\text{ERROROUT_C}}$ pin level after reset | Hi-Z (GPIO) | Hi-Z (GPIO) | Hi-Z (GPIO) | Hi-Z (GPIO) |

Note 1. In case of the debugger disconnect reset, the $\overline{\text{ERROROUT}}$ pin outputs a low level signal during reset.

32.3.3.1 Enabling Dynamic Mode

1. Select the timer output by the ECMEPCTL register (see **Section 32.4.40, ECMEPCTL — ECM Error Pulse Control Register**).
2. Initialize channel 15 in TAUD1 or OSTM1.
3. Set the $\overline{\text{ERROROUT}}$ output, and $\overline{\text{ERROROUT_C}}$ if used, to normal outputs by setting the ECMmECT (m = M/C) bit in the ECM master/checker error clear trigger register to 1.

4. Set the ECMSL0 bit in the ECM error pulse configuration register to 1 for dynamic mode.
5. Start up channel 15 in TAUD1 or OSTM1.

32.3.3.2 Disabling Dynamic Mode

1. Set the ERROROUT output, and ERROROUT_C output if used, to low level by setting the ECMmEST bit (m = M/C) in the ECM master/checker error set trigger register to 1.
2. Stop channel 15 in TAUD1 or OSTM1.
3. Clear the ECMSL0 bit in the ECM error pulse configuration register to 0 to specify non-dynamic mode.

32.3.4 Error Status

The error status is indicated by ECM master/checker error source status register 0 and ECM master/checker error source status register 1 and ECM master/checker error source status register 2. The error status is only cleared by ECM Error Source Status Clear Trigger Registers or a power on reset. In case of resets other than power on reset, the error status is kept and the error that triggered the reset can be confirmed by reading the ECM master/checker error source status register 0 and ECM master/checker error source status register 1 and ECM master/checker error source status register 2 after reset release.

32.3.5 Writing to Protected Registers

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc. An overview of the related registers is shown in **Section 32.4, Register Specification**, Overview of ECM Registers.

32.3.5.1 Protection Unlock Sequence

Write access to a write protected register is only possible within a special Protection Unlock Sequence.

1. Write the fixed value 0000 00A5_H to the protection command registers ECMPCMD1 or ECMmPCMD0.
2. Write the desired value to the registers protected by ECMPCMD1 or ECMmPCMD0.
3. Write the bit-wise inversion of desired value to the registers protected by ECMPCMD1 or ECMmPCMD0.
4. Write the desired value to the registers protected by ECMPCMD1 or ECMmPCMD0.

NOTE

Confirm successful write of the desired value to the protected register by checking that ECMPS.ECMPRERR = 0.

If an access is made to another register between step 1 to step 4 of the above sequence, the protection mechanism behaves as follows:

- When the writing is to another register in the same module, writing to the protected register fails and the EMCPS.ECMPRERR bit is set to 1.
- When writing is to a register in another module, writing to the protected register will successfully proceed to completion, if the remainder of the procedure is followed correctly.

Reading of another register while the sequence is in progress does not lead to failure.

NOTE

For sequences of ECMPCMD1 and ECMmPCMD0 registers, the status of the Protection Unlock Sequence is commonly indicated by the ECMP5.ECMPRERR flag. Therefore, it is recommended not to interleave multiple Protection Unlock Sequences of ECMPCMD1 and ECMmPCMD0.

32.3.5.2 Interrupt during Write Sequence

If an interrupt is generated during the write sequence, the protection mechanism operates as follows:

- (1) If an interrupt request is accepted during the write sequence and write access to a register of the same module is made:

The write to the write protection target register fails, and the error monitor bit is set to 1. **Figure 32.2** shows an execution example.

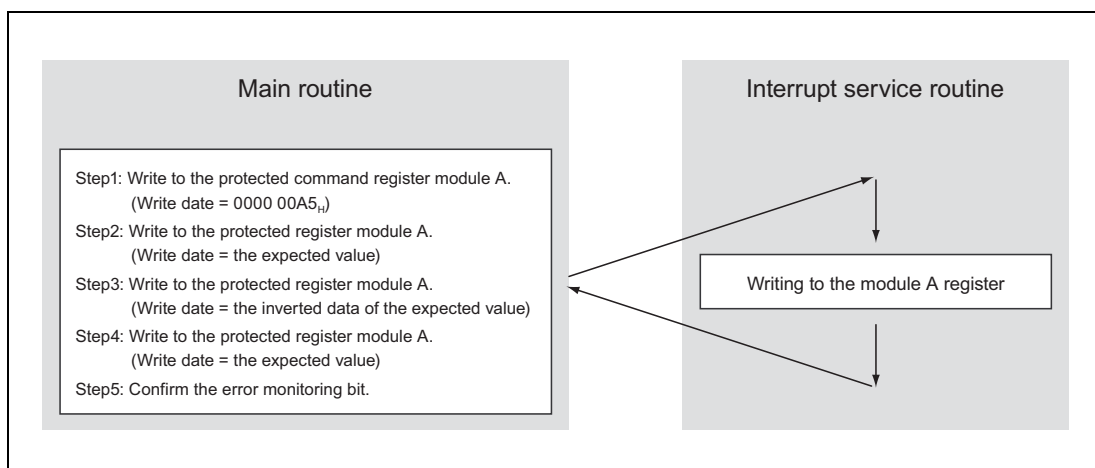


Figure 32.2 Example of Failed Write Sequence

- (2) If an interrupt request is accepted during the write sequence and write access to a register of a different module is made.

The write to the write protection register can be completed successfully. **Figure 32.3** shows an execution example.

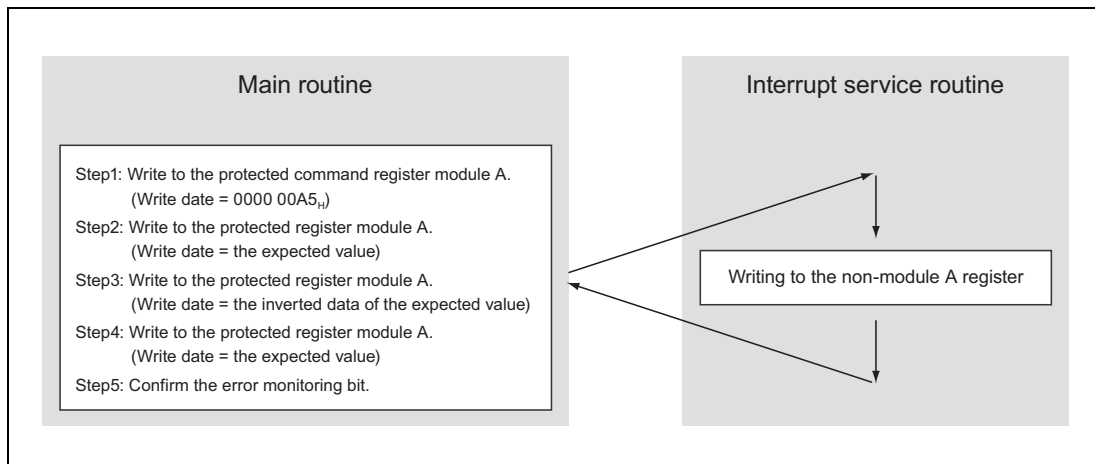


Figure 32.3 Example of Successful Write Sequence

32.3.5.3 Emulation Break during Write Sequence

If an emulation break occurs during the above write sequence, e.g. because of a breakpoint hit, the register protection is suspended until normal operation is resumed. Even if any register of the same module is accessed during the break, the write sequence is not suspended and the error monitor bit is not set to 1.

32.3.6 Timeout Function for Interrupt Processing

The delay timer incorporated in ECM can be started simultaneously with the occurrence of an interrupt request. ECM incorporates a function that generates an error signal output or Error Control Module Reset when the count value of the delay timer matches with the value of the delay timer compare register because the delay timer was not stopped during the interrupt processing. The timer counting is not stopped when a break occurs.

The counting of the delay timer always starts from 0. Configure the duration until an Error Control Module Reset or error output is generated with the settings of the delay timer compare register.

Do not set the delay timer for clock monitor upper limit/lower limit errors (Error No.8 to 15).

32.3.7 Configuration lock

The following configuration registers in ECM are protected by Protection Unlock Sequence and slave guard.

- Error Set Trigger Register (**Section 32.4.2**)
- Error Clear Trigger Register (**Section 32.4.3**)
- Error Pulse Configuration Register (**Section 32.4.8**)
- Interrupt configuration Register (**Section 32.4.9, Section 32.4.10, Section 32.4.11, Section 32.4.12, Section 32.4.13, Section 32.4.14**)
- Internal Reset configuration Register (**Section 32.4.15, Section 32.4.15, Section 32.4.17**)
- Error Mask Register (**Section 32.4.18, Section 32.4.19, Section 32.4.20**)
- Error Source Status Clear Trigger Register (**Section 32.4.21, Section 32.4.22, Section 32.4.23**)
- Pseudo Error Trigger Register (**Section 32.4.26, Section 32.4.27, Section 32.4.28**)
- Delay Timer Control Register (**Section 32.4.29**)
- Delay Timer Compare Register (**Section 32.4.31**)
- Delay Timer Configuration Register (**Section 32.4.32, Section 32.4.33, Section 32.4.34, Section 32.4.35, Section 32.4.36, Section 32.4.37**)
- Error Output Clear Invalidation Configuration Register (**Section 32.4.38**)

The detail of Protection Unlock Sequence is described in **Section 32.3.5, Writing to Protected Registers**.

Slave guard is described in **Section 31, Functional Safety**.

32.3.8 Masking of error clear trigger registers

The active error output status must be cleared by software via the Error clear trigger registers (ECMMECLR/ECMCECLR). A minimum activation time of the error output is achieved by the Error

output clear invalidation counter. This counter is (re)started each time a new error event is triggered at the ECM. It counts up from 0000_H to FFFF_H. Error output clear by software is not possible unless this counter reaches the compare value configured in the ECMEOCCFG register. If Error output clear invalidation counter is still running, Error output clear is masked and Error output clear request by software is ignored.

32.4 Register Specification

32.4.1 List of Registers

ECM consists of three address areas: common part, ECM master, and ECM checker.

The following shows the register map of the ECM master and checker registers.

Table 32.12 Address List of ECM Master and Checker Registers

| Address | Register Symbol | Register Name | R/W | Access Width | Value after reset | Protection Registers | |
|-------------------------------|-----------------|--|-----|--------------|------------------------|---|-----------------|
| | | | | | | Command Register | Status Register |
| ECM Master Registers | | | | | | <ECMM_base: FFD6 0000_H> | |
| <ECMM_base> | ECMMESET | ECM master error set trigger register | W | 8 | 00 _H | ECMMPCMD0 | ECMPS |
| <ECMM_base> + 04 _H | ECMMECLR | ECM master error clear trigger register | W | 8 | 00 _H | ECMMPCMD0 | ECMPS |
| <ECMM_base> + 08 _H | ECMMESSTR0 | ECM master error source status register 0 | R | 32 | 0000 0000 _H | — | — |
| <ECMM_base> + 0C _H | ECMMESSTR1 | ECM master error source status register 1 | R | 32 | 0000 0000 _H | — | — |
| <ECMM_base> + 10 _H | ECMMESSTR2 | ECM master error source status register 2 | R | 32 | 0000 0000 _H | — | — |
| <ECMM_base> + 14 _H | ECMMPCMD0 | ECM master protection command register | W | 32 | Undefined | — | — |
| ECM Checker Registers | | | | | | <ECMC_base: FFD6 1000_H> | |
| <ECMC_base> | ECMCESET | ECM checker error set trigger register | W | 8 | 00 _H | ECMCPCMD0 | ECMPS |
| <ECMC_base> + 04 _H | ECMCECLR | ECM checker error clear trigger register | W | 8 | 00 _H | ECMCPCMD0 | ECMPS |
| <ECMC_base> + 08 _H | ECMCESSTR0 | ECM checker error source status register 0 | R | 32 | 0000 0000 _H | — | — |
| <ECMC_base> + 0C _H | ECMCESSTR1 | ECM checker error source status register 1 | R | 32 | 0000 0000 _H | — | — |
| <ECMC_base> + 10 _H | ECMCESSTR2 | ECM checker error source status register 2 | R | 32 | 0000 0000 _H | — | — |
| <ECMC_base> + 14 _H | ECMCPCMD0 | ECM checker protection command register | W | 32 | Undefined | — | — |

The following shows the register map of the ECM common part.

Table 32.13 Address List of ECM Common Registers (1/2)

| <ECM_base: FFD6 2000 _H > | | | | | | | |
|-------------------------------------|-----------------|---|-----|--------------|------------------------|----------------------|-----------------|
| Address | Register Symbol | Register Name | R/W | Access Width | Value after reset | Protection Registers | |
| | | | | | | Command Register | Status Register |
| <ECM_base> | ECMEPCFG | ECM error pulse configuration register | R/W | 8 | 00 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 04 _H | ECMMICFG0 | ECM maskable interrupt configuration register 0 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 08 _H | ECMMICFG1 | ECM maskable interrupt configuration register 1 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 0C _H | ECMMICFG2 | ECM maskable interrupt configuration register 2 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 10 _H | ECMNMICFG0 | ECM non-maskable interrupt configuration register 0 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 14 _H | ECMNMICFG1 | ECM non-maskable interrupt configuration register 1 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 18 _H | ECMNMICFG2 | ECM non-maskable interrupt configuration register 2 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 1C _H | ECMIRCFG0 | ECM internal reset configuration register 0 | R/W | 32 | 0000 0001 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 20 _H | ECMIRCFG1 | ECM internal reset configuration register 1 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 24 _H | ECMIRCFG2 | ECM internal reset configuration register 2 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 28 _H | ECMEMK0 | ECM error mask register 0 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 2C _H | ECMEMK1 | ECM error mask register 1 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 30 _H | ECMEMK2 | ECM error mask register 2 | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 34 _H | ECMESSTC0 | ECM error source status clear register 0 | W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 38 _H | ECMESSTC1 | ECM error source status clear register 1 | W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 3C _H | ECMESSTC2 | ECM error source status clear register 2 | W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 40 _H | ECMPCMD1 | ECM protection command register | W | 32 | Undefined | — | — |
| <ECM_base> + 44 _H | ECMPS | ECM protection status register | R | 8 | 00 _H | — | — |
| <ECM_base> + 48 _H | ECMPE0 | ECM pseudo error trigger register 0 | W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 4C _H | ECMPE1 | ECM pseudo error trigger register 1 | W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 50 _H | ECMPE2 | ECM pseudo error trigger register 2 | W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 54 _H | ECMDTMCTL | ECM delay timer control register | R/W | 8 | 00 _H | ECMPCMD1 | ECMPS |
| <ECM_base> + 58 _H | ECMDTMR | ECM delay timer register | R | 16 | 0000 _H | — | — |
| <ECM_base> + 5C _H | ECMDTMCMP | ECM delay timer compare register | R/W | 32 | 0000 0000 _H | ECMPCMD1 | ECMPS |

Table 32.13 Address List of ECM Common Registers (2/2)

| <ECM_base: FFD6 2000 _H > | | | | | | | |
|-------------------------------------|-----------------|--|-----|--------------|------------------------|----------------------|-----------------|
| Address | Register Symbol | Register Name | R/W | Access Width | Value after reset | Protection Registers | |
| | | | | | | Command Register | Status Register |
| <ECM_base> + 60 _H | ECMDTMCFG0 | ECM delay timer configuration register 0 | R/W | 32 | 0000 0000 _H | ECMPS | ECMPS |
| <ECM_base> + 64 _H | ECMDTMCFG1 | ECM delay timer configuration register 1 | R/W | 32 | 0000 0000 _H | ECMPS | ECMPS |
| <ECM_base> + 68 _H | ECMDTMCFG2 | ECM delay timer configuration register 2 | R/W | 32 | 0000 0000 _H | ECMPS | ECMPS |
| <ECM_base> + 6C _H | ECMDTMCFG3 | ECM delay timer configuration register 3 | R/W | 32 | 0000 0000 _H | ECMPS | ECMPS |
| <ECM_base> + 70 _H | ECMDTMCFG4 | ECM delay timer configuration register 4 | R/W | 32 | 0000 0000 _H | ECMPS | ECMPS |
| <ECM_base> + 74 _H | ECMDTMCFG5 | ECM delay timer configuration register 5 | R/W | 32 | 0000 0000 _H | ECMPS | ECMPS |
| <ECM_base> + 78 _H | ECMEOCCFG | ECM error output clear invalidation configuration register | R/W | 32 | 0000 0000 _H | ECMPS | ECMPS |
| <ECM_base> + 7C _H | ECMPEM | ECM pseudo error mask register | R/W | 32 | 0000 0000 _H | — | — |
| FFD6 3000 _H | ECMEPCTL | ECM error pulse control register | R/W | 8 | 00 _H | — | — |

The ECM registers are assigned to three address areas: the address area for ECM common registers, the address area for ECM master registers, and the address area for ECM checker registers. The address area for ECM common registers is used for both master and checker registers. Writing to this common area is performed by master and checker registers simultaneously. Reading from the common area reads the master register's value. The ECM master and checker registers can be written separately.

ECM Register reset conditions are shown in **Table 32.14** and **Table 32.15**.

Table 32.14 Reset Condition of ECM Master and Checker Registers (1/2)

| Register Symbol | Register Name | Reset condition | | | |
|------------------------------|---|-----------------|---------------|---------------|--------------------|
| | | Power On Reset | System Reset1 | System Reset2 | Application Reset1 |
| ECM Master Registers | | | | | |
| ECMMESET | ECM master error set trigger register | √ | √ | √ | √ |
| ECMMECLR | ECM master error clear trigger register | √ | √ | √ | √ |
| ECMMESSTR0 | ECM master error source status register 0 | √*1 | — | — | — |
| ECMMESSTR1 | ECM master error source status register 1 | √*1 | — | — | — |
| ECMMESSTR2 | ECM master error source status register 2 | √*1 | — | — | — |
| ECMMPCMD0 | ECM master protection command register | √ | √ | √ | √ |
| ECM Checker Registers | | | | | |
| ECMCESET | ECM checker error set trigger register | √ | √ | √ | √ |
| ECMCECLR | ECM checker error clear trigger register | √ | √ | √ | √ |

Table 32.14 Reset Condition of ECM Master and Checker Registers (2/2)

| Register Symbol | Register Name | Reset condition | | | |
|-----------------|--|-----------------|---------------|---------------|--------------------|
| | | Power On Reset | System Reset1 | System Reset2 | Application Reset1 |
| ECMCESSTR0 | ECM checker error source status register 0 | √*1 | — | — | — |
| ECMCESSTR1 | ECM checker error source status register 1 | √*1 | — | — | — |
| ECMCESSTR2 | ECM checker error source status register 2 | √*1 | — | — | — |
| ECMCPCMD0 | ECM checker protection command register | √ | √ | √ | √ |

Note 1. Except Debug Initiated Reset.

Table 32.15 Reset Condition of ECM Common Registers (1/2)

| Register Symbol | Register Name | Reset condition | | | |
|-----------------|---|-----------------|---------------|---------------|--------------------|
| | | Power On Reset | System Reset1 | System Reset2 | Application Reset1 |
| ECMEPCFG | ECM error pulse configuration register | √ | √ | √ | √ |
| ECMMICFG0 | ECM maskable interrupt configuration register 0 | √ | √ | √ | √ |
| ECMMICFG1 | ECM maskable interrupt configuration register 1 | √ | √ | √ | √ |
| ECMMICFG2 | ECM maskable interrupt configuration register 2 | √ | √ | √ | √ |
| ECMNMICFG0 | ECM non-maskable interrupt configuration register 0 | √ | √ | √ | √ |
| ECMNMICFG1 | ECM non-maskable interrupt configuration register 1 | √ | √ | √ | √ |
| ECMNMICFG2 | ECM non-maskable interrupt configuration register 2 | √ | √ | √ | √ |
| ECMIRCFG0 | ECM internal reset configuration register 0 | √ | √ | √ | √ |
| ECMIRCFG1 | ECM internal reset configuration register 1 | √ | √ | √ | √ |
| ECMIRCFG2 | ECM internal reset configuration register 2 | √ | √ | √ | √ |
| ECMEMK0 | ECM error mask register 0 | √ | √ | √ | √ |
| ECMEMK1 | ECM error mask register 1 | √ | √ | √ | √ |
| ECMEMK2 | ECM error mask register 2 | √ | √ | √ | √ |
| ECMESSTC0 | ECM error source status clear register 0 | √ | √ | √ | √ |
| ECMESSTC1 | ECM error source status clear register 1 | √ | √ | √ | √ |
| ECMESSTC2 | ECM error source status clear register 2 | √ | √ | √ | √ |
| ECMPCMD1 | ECM protection command register | √ | √ | √ | √ |
| ECMPS | ECM protection status register | √ | √ | √ | √ |
| ECMPE0 | ECM pseudo error trigger register 0 | √ | √ | √ | √ |
| ECMPE1 | ECM pseudo error trigger register 1 | √ | √ | √ | √ |

Table 32.15 Reset Condition of ECM Common Registers (2/2)

| Register Symbol | Register Name | Reset condition | | | |
|-----------------|--|-----------------|---------------|---------------|--------------------|
| | | Power On Reset | System Reset1 | System Reset2 | Application Reset1 |
| ECMPE2 | ECM pseudo error trigger register 2 | √ | √ | √ | √ |
| ECMDTMCTL | ECM delay timer control register | √ | √ | √ | √ |
| ECMDTMR | ECM delay timer register | √ | √ | √ | √ |
| ECMDTMCMP | ECM delay timer compare register | √ | √ | √ | √ |
| ECMDTMCFG0 | ECM delay timer configuration register 0 | √ | √ | √ | √ |
| ECMDTMCFG1 | ECM delay timer configuration register 1 | √ | √ | √ | √ |
| ECMDTMCFG2 | ECM delay timer configuration register 2 | √ | √ | √ | √ |
| ECMDTMCFG3 | ECM delay timer configuration register 3 | √ | √ | √ | √ |
| ECMDTMCFG4 | ECM delay timer configuration register 4 | √ | √ | √ | √ |
| ECMDTMCFG5 | ECM delay timer configuration register 5 | √ | √ | √ | √ |
| ECMEOCCFG | ECM error output clear invalidation configuration register | √ | √ | √ | — |
| ECMPEM | ECM pseudo error mask register | √ | √ | √ | √ |

32.4.2 ECMmESET — ECM Master/Checker Error Set Trigger Register (m = M/C)

The ECM master/checker error set trigger register is for setting the error signal from the error pin to the low level. When the ECMmEST bit is set to 1, the error pin outputs the low level. The output cannot be masked.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

This register is always read as 00_H.

Access: This register can be written in 8-bit units.

Address: <ECMM_base>
<ECMC_base>

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | ECMmEST |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 32.16 ECMmESET Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ECMmEST | Error set trigger bit 0: Writing 0 is invalid 1: Sets the output level from the error pin to the active (low) level. |

CAUTIONS

Setting or clearing the error output from the $\overline{\text{ERROROUT}}$ pin via the ECMmESET or ECMmECLR register will set the ECMmSSE228 bit of the ECMmESSTR2 register (ECM compare error). Therefore, the ECMmESET register has to be set in the following sequence.

1. Set the MSKM bit and MSKC bit of the ECMPEM register to “masked”.
2. Set the ECMmEST bit in the ECMmESET register.
3. Wait until $\overline{\text{ERROROUT}}/\overline{\text{ERROROUT_C}}$ becomes low by checking that the ECMmSSE231 bit of the ECMmESSTR2 register is “0”.
4. Set the MSKM bit and MSKC bit of the ECMPEM register to “not masked”.

32.4.3 ECMmECLR — ECM Master/Checker Error Clear Trigger Register (m = M/C)

The ECM master/checker error clear trigger registers are for setting the error signal from the error pin(s) to the high level (toggle). When the ECMmECT bit is set to 1, the error pin outputs the high level (toggle) as long as there are no other sources that set the error pin to the low level.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence. This register is always read as 00_H.

Access: This register can be written in 8-bit units.

Address: <ECMM_base> + 04_H
<ECMC_base> + 04_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---------|
| | — | — | — | — | — | — | — | ECMmECT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

Table 32.17 ECMmECLR Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ECMmECT | Error clear trigger bit 0: Writing 0 is invalid 1: Sets the output level from the error pin to the inactive (high) level. |

CAUTIONS

Clearing of the error pin is only possible if all errors, not masked by ECMEMK0/1/2, are cleared beforehand.

Setting or clearing the error output via the ECMmECLR register will generate an error. Therefore, the following has to be set in advance.

1. Set the MSKM bit and MSKC bit of the ECMPPEM register to “masked”.
2. Set the ECMmECT bit in the ECMmECLR register.
3. Wait until $\overline{\text{ERROROUT}}/\overline{\text{ERROROUT_C}}$ becomes high by reading ECMmSSE231 bit of ECMmESSTR2 register 30 times. After that, check that the ECMmSSE231 bit of the ECMmESSTR2 register is “1”. If the ECMmSSE231 bit of the ECMmESSTR2 register is not “1”, a new error may have occurred.
4. Set the MSKM bit and MSKC bit of the ECMPPEM register to “not masked”.

Note: This procedure is applicable when the ECMEOCCFG register is not set.

NOTE

If the “output clear invalidation counter”, which is configured by ECMEOCCFG, the Error Output clear function is masked and Error Output clear requests are ignored. Error Output clear function is executed, after the Error Pin Low Time Counter has expired and the Error Output clear register is written.

32.4.4 ECMmESSTR0 — ECM Master/Checker Error Source Status Register 0 (m = M/C)

The ECM master/checker error source status register 0 is a read-only register. This register represents the status of individual internal error sources which are independent from the settings of the error masks. The status can be cleared only by software and power on reset (except debug initiated reset). Other resets than power on reset will not affect the status.

Access: This register can be read in 32-bit units.

Address: <ECMM_base> + 08_H
<ECMC_base> + 08_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | ECMmS SE031 | ECMmS SE030 | ECMmS SE029 | ECMmS SE028 | ECMmS SE027 | ECMmS SE026 | ECMmS SE025 | ECMmS SE024 | ECMmS SE023 | ECMmS SE022 | ECMmS SE021 | ECMmS SE020 | ECMmS SE019 | ECMmS SE018 | ECMmS SE017 | ECMmS SE016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMmS SE015 | ECMmS SE014 | ECMmS SE013 | ECMmS SE012 | ECMmS SE011 | ECMmS SE010 | ECMmS SE009 | ECMmS SE008 | ECMmS SE007 | ECMmS SE006 | ECMmS SE005 | ECMmS SE004 | ECMmS SE003 | ECMmS SE002 | ECMmS SE001 | ECMmS SE000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 32.18 ECMmESSTR0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------------------|--|
| 31 to 0 | ECMmSSE031 to ECMmSSE000 | Error source status bit ECMmSSE031 to ECMmSSE000 correspond to error sources 31 to 0. 0: Error not occurred 1: Error occurred |

32.4.5 ECMmESSTR1 — ECM Master/Checker Error Source Status Register 1 (m = M/C)

The ECM master/checker error source status register 1 is a read-only register. This register represents the status of individual internal error sources which are independent from the settings of the error masks. The status can be cleared only by software and power on reset (except debug initiated reset). Other resets than power on reset will not affect the status.

Access: This register can be read in 32-bit units.

Address: <ECMM_base> + 0C_H
<ECMC_base> + 0C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | ECMmSSE131 | ECMmSSE130 | ECMmSSE129 | ECMmSSE128 | ECMmSSE127 | ECMmSSE126 | ECMmSSE125 | ECMmSSE124 | ECMmSSE123 | ECMmSSE122 | ECMmSSE121 | ECMmSSE120 | ECMmSSE119 | ECMmSSE118 | ECMmSSE117 | ECMmSSE116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMmSSE115 | ECMmSSE114 | ECMmSSE113 | ECMmSSE112 | ECMmSSE111 | ECMmSSE110 | ECMmSSE109 | ECMmSSE108 | ECMmSSE107 | ECMmSSE106 | ECMmSSE105 | ECMmSSE104 | ECMmSSE103 | ECMmSSE102 | ECMmSSE101 | ECMmSSE100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 32.19 ECMmESSTR1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------------|---|
| 31 to 0 | ECMmSSE131 to ECMmSSE100 | Error source status bit ECMmSSE131 to ECMmSSE100 correspond to error sources 63 to 32. 0: Error not occurred 1: Error occurred |

32.4.6 ECMmESSTR2 — ECM Master/Checker Error Source Status Register 2 (m = M/C)

The ECM master/checker error source status register 2 is a read-only register.

This register represents the status of individual internal error sources which are independent from the settings of the error masks. The status can be cleared only by software and power on reset (except debug initiated reset). Other resets than power on reset will not affect the status. As for bit 31, it does not have a reset and it reflects the level of $\overline{\text{ERROROUT}}/\overline{\text{ERROROUT_C}}$ pin.

Access: This register can be read in 32-bit units.

Address: <ECMM_base> + 10_H
<ECMC_base> + 10_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | ECMmSSE231 | ECMmSSE230 | ECMmSSE229 | ECMmSSE228 | ECMmSSE227 | ECMmSSE226 | ECMmSSE225 | ECMmSSE224 | ECMmSSE223 | ECMmSSE222 | ECMmSSE221 | ECMmSSE220 | ECMmSSE219 | ECMmSSE218 | ECMmSSE217 | ECMmSSE216 |
| Value after reset | — ¹ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMmSSE215 | ECMmSSE214 | ECMmSSE213 | ECMmSSE212 | ECMmSSE211 | ECMmSSE210 | ECMmSSE209 | ECMmSSE208 | ECMmSSE207 | ECMmSSE206 | ECMmSSE205 | ECMmSSE204 | ECMmSSE203 | ECMmSSE202 | ECMmSSE201 | ECMmSSE200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Note 1. Depends on $\overline{\text{ERROROUT}}$ pin state (m=M) or $\overline{\text{ERROROUT_C}}$ pin state (m=C). When using $\overline{\text{ERROROUT_C}}$ pin, set the alternative output level loopback function.

Table 32.20 ECMmESSTR2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------------|---|
| 31 | ECMmSSE231 | The status of the $\overline{\text{ERROROUT}}$ pin 0: $\overline{\text{ERROROUT}}$ is low level 1: $\overline{\text{ERROROUT}}$ is high level |
| 30 | ECMmSSE230 | The status of the ECMmESET writing 0: No Error 1: Error is set by ECMmESET |
| 29 | ECMmSSE229 | The status of the delay timer overflow 0: No overflow 1: Overflow |
| 28 to 0 | ECMmSSE228 to ECMmSSE200 | Error source status bit ECMmSSE228 to ECMmSSE200 correspond to error sources 92 to 64. 0: Error not occurred 1: Error occurred |

NOTE

After Field-BIST executes, regardless of whether it is an intended execution or not, ECMmESSTR2.ECMmSSE227 will be always "1". In the startup (after reset release), Field-BIST will be always executed and "1" will be captured in ECMmESSTR2.bit 27.

Therefore, confirm that ECMmESSTR2.ECMmSSE227 is 1 and then clear this bit by software before user operation starts.

32.4.7 ECMmPCMD0 — ECM Master/Checker Protection Command Register (m = M/C)

The ECM master/checker protection command register is a write-only register and can be written in 32-bit units. Refer to **Section 32.4.1, List of Registers**, for the protected registers.

Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence. The value after reset is undefined.

Access: This register can be written 32-bit units.

Address: <ECMM_base> + 14_H
<ECMC_base> + 14_H

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | Undefined | | | | | | | | | | | | | | | |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | ECMmREG0[7:0] | | | | | | | |
| Value after reset | Undefined | | | | | | | | | | | | | | | |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.21 ECMmPCMD0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|--|
| 31 to 8 | Reserved | When writing, write 0. |
| 7 to 0 | ECMmREG0 [7:0] | Protection command that enables writing to write protected ECMm registers. |

32.4.8 ECMEPCFG — ECM Error Pulse Configuration Register

The ECM error pulse configuration register sets the error output operation for the $\overline{\text{ERROROUT}}$ pin. The ECM error pulse configuration register is a read/write register. You have to follow Protection Unlock Sequence for writing data to this register.

Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 8-bit units.

Address: <ECM_base>

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|--------|
| | — | — | — | — | — | — | — | ECMSL0 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 32.22 ECMEPCFG Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | ECMSL0 | Error pin output operation configuration bit Error output operation setting for the error pin 0: Non-dynamic mode 1: Dynamic mode |

In Dynamic mode the timer output channel 15 in TAUD1 or OSTM1 determines the output wave of the $\overline{\text{ERROROUT}}$ / $\overline{\text{ERROROUT_C}}$ pins in case of no error.

32.4.9 ECMMICFG0 — ECM Maskable Interrupt Configuration Register 0

The ECM maskable interrupt configuration register 0 is used to set the generation of the INTECM interrupts (EI level interrupts). The generation of EI level interrupts in response to errors can be enabled.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 04_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | ECMMI E031 | ECMMI E030 | ECMMI E029 | ECMMI E028 | ECMMI E027 | ECMMI E026 | ECMMI E025 | ECMMI E024 | ECMMI E023 | ECMMI E022 | ECMMI E021 | ECMMI E020 | ECMMI E019 | ECMMI E018 | ECMMI E017 | ECMMI E016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | ECMMI E015 | ECMMI E014 | ECMMI E013 | ECMMI E012 | ECMMI E011 | ECMMI E010 | ECMMI E009 | ECMMI E008 | ECMMI E007 | ECMMI E006 | ECMMI E005 | ECMMI E004 | ECMMI E003 | ECMMI E002 | ECMMI E001 | ECMMI E000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.23 ECMMICFG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------------|---|
| 32 to 0 | ECMMIE031 to ECMMIE000 | ECM maskable interrupt generation control bit ECMMIE031 to ECMMIE000 correspond to error sources 31 to 0. 0: Interrupt generation disabled 1: Interrupt generation enabled |

32.4.10 ECMMICFG1 — ECM Maskable Interrupt Configuration Register 1

The ECM maskable interrupt configuration register 1 is used to set the generation of the INTECM interrupts (EI level interrupts). The generation of EI level interrupts in response to errors can be enabled.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 08_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | ECMMI E131 | ECMMI E130 | ECMMI E129 | ECMMI E128 | ECMMI E127 | ECMMI E126 | ECMMI E125 | ECMMI E124 | ECMMI E123 | ECMMI E122 | ECMMI E121 | ECMMI E120 | ECMMI E119 | ECMMI E118 | ECMMI E117 | ECMMI E116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | ECMMI E115 | ECMMI E114 | ECMMI E113 | ECMMI E112 | ECMMI E111 | ECMMI E110 | ECMMI E109 | ECMMI E108 | ECMMI E107 | ECMMI E106 | ECMMI E105 | ECMMI E104 | ECMMI E103 | ECMMI E102 | ECMMI E101 | ECMMI E100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.24 ECMMICFG1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------------|--|
| 31 to 0 | ECMMIE131 to ECMMIE100 | ECM maskable interrupt generation control bit ECMMIE131 to ECMMIE100 correspond to error sources 63 to 32. 0: Interrupt generation disabled 1: Interrupt generation enabled |

32.4.11 ECMMICFG2 — ECM Maskable Interrupt Configuration Register 2

The ECM maskable interrupt configuration register 2 is used to set the generation of the INTECM interrupts (EI level interrupts). The generation of EI level interrupts in response to errors can be enabled. You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 0C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | — | — | — | ECMMI E228 | ECMMI E227 | ECMMI E226 | ECMMI E225 | ECMMI E224 | ECMMI E223 | ECMMI E222 | ECMMI E221 | ECMMI E220 | ECMMI E219 | ECMMI E218 | ECMMI E217 | ECMMI E216 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMMI E215 | ECMMI E214 | ECMMI E213 | ECMMI E212 | ECMMI E211 | ECMMI E210 | ECMMI E209 | ECMMI E208 | ECMMI E207 | ECMMI E206 | ECMMI E205 | ECMMI E204 | ECMMI E203 | ECMMI E202 | ECMMI E201 | ECMMI E200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.25 ECMMICFG2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------------|--|
| 31 to 29 | Reserved | When writing, write the value after reset. |
| 28 to 0 | ECMMIE228 to ECMMIE200 | ECM maskable interrupt generation control bit ECMMIE228 to ECMMIE200 correspond to error sources 92 to 64. 0: Interrupt generation disabled 1: Interrupt generation enabled |

32.4.12 ECMNMICFG0 — ECM Non-maskable Interrupt Configuration Register 0

The ECM Non-maskable interrupt configuration register 0 is used to set the generation of INTECMNMI interrupts.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 10_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | ECMNMIE031 | ECMNMIE030 | ECMNMIE029 | ECMNMIE028 | ECMNMIE027 | ECMNMIE026 | ECMNMIE025 | ECMNMIE024 | ECMNMIE023 | ECMNMIE022 | ECMNMIE021 | ECMNMIE020 | ECMNMIE019 | ECMNMIE018 | ECMNMIE017 | ECMNMIE016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMNMIE015 | ECMNMIE014 | ECMNMIE013 | ECMNMIE012 | ECMNMIE011 | ECMNMIE010 | ECMNMIE009 | ECMNMIE008 | ECMNMIE007 | ECMNMIE006 | ECMNMIE005 | ECMNMIE004 | ECMNMIE003 | ECMNMIE002 | ECMNMIE001 | ECMNMIE000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.26 ECMNMICFG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------------------|---|
| 31 to 0 | ECMNMIE031 to ECMNMIE000 | ECM Non-maskable interrupt generation control bit ECMNMIE031 to ECMNMIE000 correspond to error sources 31 to 0. 0: Interrupt generation disabled 1: Interrupt generation enabled |

32.4.13 ECMNMICFG1 — ECM Non-maskable Interrupt Configuration Register 1

The ECM Non-maskable interrupt configuration register 1 is used to set the generation of INTECMNMI interrupts.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 14_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | ECMNMIE131 | ECMNMIE130 | ECMNMIE129 | ECMNMIE128 | ECMNMIE127 | ECMNMIE126 | ECMNMIE125 | ECMNMIE124 | ECMNMIE123 | ECMNMIE122 | ECMNMIE121 | ECMNMIE120 | ECMNMIE119 | ECMNMIE118 | ECMNMIE117 | ECMNMIE116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMNMIE115 | ECMNMIE114 | ECMNMIE113 | ECMNMIE112 | ECMNMIE111 | ECMNMIE110 | ECMNMIE109 | ECMNMIE108 | ECMNMIE107 | ECMNMIE106 | ECMNMIE105 | ECMNMIE104 | ECMNMIE103 | ECMNMIE102 | ECMNMIE101 | ECMNMIE100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.27 ECMNMICFG1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------------|--|
| 31 to 0 | ECMNMIE131 to ECMNMIE100 | ECM Non-maskable interrupt generation control bit ECMNMIE131 to ECMNMIE100 correspond to error sources 63 to 32. 0: Interrupt generation disabled 1: Interrupt generation enabled |

32.4.14 ECMNMICFG2 — ECM Non-maskable Interrupt Configuration Register 2

The ECM Non-maskable interrupt configuration register 2 is used to set the generation of INTECMNMI interrupts.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 18_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | — | — | — | ECMNMIE228 | ECMNMIE227 | ECMNMIE226 | ECMNMIE225 | ECMNMIE224 | ECMNMIE223 | ECMNMIE222 | ECMNMIE221 | ECMNMIE220 | ECMNMIE219 | ECMNMIE218 | ECMNMIE217 | ECMNMIE216 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMNMIE215 | ECMNMIE214 | ECMNMIE213 | ECMNMIE212 | ECMNMIE211 | ECMNMIE210 | ECMNMIE209 | ECMNMIE208 | ECMNMIE207 | ECMNMIE206 | ECMNMIE205 | ECMNMIE204 | ECMNMIE203 | ECMNMIE202 | ECMNMIE201 | ECMNMIE200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.28 ECMNMICFG2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------------------|--|
| 31 to 29 | Reserved | When writing, write the value after reset. |
| 28 to 0 | ECMNMIE228 to ECMNMIE200 | ECM Non-maskable interrupt generation control bit ECMNMIE228 to ECMNMIE200 correspond to error sources 92 to 64. 0: Interrupt generation disabled 1: Interrupt generation enabled |

32.4.15 ECMIRCFG0 — ECM Internal Reset Configuration Register 0

The ECM internal reset configuration register 0 is used to set the generation of Error Control Module Reset in response to internal errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 1C_H

Value after reset: 0000 0001_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | ECMIR E031 | ECMIR E030 | ECMIR E029 | ECMIR E028 | ECMIR E027 | ECMIR E026 | ECMIR E025 | ECMIR E024 | ECMIR E023 | ECMIR E022 | ECMIR E021 | ECMIR E020 | ECMIR E019 | ECMIR E018 | ECMIR E017 | ECMIR E016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMIR E015 | ECMIR E014 | ECMIR E013 | ECMIR E012 | ECMIR E011 | ECMIR E010 | ECMIR E009 | ECMIR E008 | ECMIR E007 | ECMIR E006 | ECMIR E005 | ECMIR E004 | ECMIR E003 | ECMIR E002 | ECMIR E001 | ECMIR E000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.29 ECMIRCFG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 31 to 0 | ECMIRE031 to ECMIRE000 | ECM internal reset generation control bit ECMIRE031 to ECMIRE000 correspond to error sources 31 to 0. 0: Error Control Module Reset generation disabled 1: Error Control Module Reset generation enabled |

NOTE

Only the watchdog timer error is enabled by the default setting of Error Control Module Reset. Please refer to **Table 32.9, List of Error Inputs**. The default setting is useful to support the default start mode of WDTA. The start mode is selectable by flash option setting. The WDTA provides two modes for the counter start after reset release:

- Software trigger start mode: The counter value remains 0000_H after reset release. The counter is started with the first WDTA trigger. The first trigger can occur any time after reset release.
- Default start mode: The counter starts automatically after reset release. The first trigger must occur before the counter overflows.

32.4.16 ECMIRCFG1 — ECM Internal Reset Configuration Register 1

The ECM internal reset configuration register 1 is used to set the generation of Error Control Module Reset in response to internal errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 20_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | ECMIR E131 | ECMIR E130 | ECMIR E129 | ECMIR E128 | ECMIR E127 | ECMIR E126 | ECMIR E125 | ECMIR E124 | ECMIR E123 | ECMIR E122 | ECMIR E121 | ECMIR E120 | ECMIR E119 | ECMIR E118 | ECMIR E117 | ECMIR E116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMIR E115 | ECMIR E114 | ECMIR E113 | ECMIR E112 | ECMIR E111 | ECMIR E110 | ECMIR E109 | ECMIR E108 | ECMIR E107 | ECMIR E106 | ECMIR E105 | ECMIR E104 | ECMIR E103 | ECMIR E102 | ECMIR E101 | ECMIR E100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.30 ECMIRCFG1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|--|
| 31 to 0 | ECMIRE131 to ECMIRE100 | ECM Internal reset generation control bit ECMIRE131 to ECMIRE100 correspond to error sources 63 to 32. 0: Error Control Module Reset generation disabled 1: Error Control Module Reset generation enabled |

32.4.17 ECMIRCFG2 — ECM Internal Reset Configuration Register 2

The ECM internal reset configuration register 2 is used to set the generation of Error Control Module Reset in response to internal errors. You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 24_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | — | — | ECMIR E229 | ECMIR E228 | ECMIR E227 | ECMIR E226 | ECMIR E225 | ECMIR E224 | ECMIR E223 | ECMIR E222 | ECMIR E221 | ECMIR E220 | ECMIR E219 | ECMIR E218 | ECMIR E217 | ECMIR E216 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMIR E215 | ECMIR E214 | ECMIR E213 | ECMIR E212 | ECMIR E211 | ECMIR E210 | ECMIR E209 | ECMIR E208 | ECMIR E207 | ECMIR E206 | ECMIR E205 | ECMIR E204 | ECMIR E203 | ECMIR E202 | ECMIR E201 | ECMIR E200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.31 ECMIRCFG2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------------------|--|
| 31, 30 | Reserved | When writing, write the value after reset. |
| 29 | ECMIRE229 | ECM internal reset control bit. Corresponds to delay timer overflow. 0: Error Control Module Reset generation disabled. 1: Error Control Module Reset generation enabled. |
| 28 to 0 | ECMIRE228 to ECMIRE200 | ECM internal reset generation control bit ECMIRE228 to ECMIRE200 correspond to error sources 92 to 64. 0: Error Control Module Reset generation disabled 1: Error Control Module Reset generation enabled |

32.4.18 ECMEMK0 — ECM Error Mask Register 0

The ECM error mask register 0 is used to mask the individual error sources of the error pin output(s).

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 28_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | ECMEMK031 | ECMEMK030 | ECMEMK029 | ECMEMK028 | ECMEMK027 | ECMEMK026 | ECMEMK025 | ECMEMK024 | ECMEMK023 | ECMEMK022 | ECMEMK021 | ECMEMK020 | ECMEMK019 | ECMEMK018 | ECMEMK017 | ECMEMK016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMEMK015 | ECMEMK014 | ECMEMK013 | ECMEMK012 | ECMEMK011 | ECMEMK010 | ECMEMK009 | ECMEMK008 | ECMEMK007 | ECMEMK006 | ECMEMK005 | ECMEMK004 | ECMEMK003 | ECMEMK002 | ECMEMK001 | ECMEMK000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.32 ECMEMK0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|---|
| 31 to 0 | ECMEMK031 to ECMEMK000 | ECM error output signal mask control bit ECMEMK031 to ECMEMK000 correspond to error sources 31 to 0. 0: Error signal output not masked 1: Error signal output masked |

NOTE

If an error flag is set but masked, clearing the mask will set the $\overline{\text{ERROROUT}}/\overline{\text{ERROROUT_C}}$ to low level.

This happens independently from the time of the error occurrence. In other words, the flag is evaluated statically.

32.4.19 ECMEMK1 — ECM Error Mask Register 1

The ECM error mask register 1 is used to mask the individual error sources of the error pin output(s).

You have to follow Protection Unlock Sequence for writing data to this register.

Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 2C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | ECMEMK131 | ECMEMK130 | ECMEMK129 | ECMEMK128 | ECMEMK127 | ECMEMK126 | ECMEMK125 | ECMEMK124 | ECMEMK123 | ECMEMK122 | ECMEMK121 | ECMEMK120 | ECMEMK119 | ECMEMK118 | ECMEMK117 | ECMEMK116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMEMK115 | ECMEMK114 | ECMEMK113 | ECMEMK112 | ECMEMK111 | ECMEMK110 | ECMEMK109 | ECMEMK108 | ECMEMK107 | ECMEMK106 | ECMEMK105 | ECMEMK104 | ECMEMK103 | ECMEMK102 | ECMEMK101 | ECMEMK100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.33 ECMEMK1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|--|
| 31 to 0 | ECMEMK131 to ECMEMK100 | ECM error output signal mask control bit ECMEMK131 to ECMEMK100 correspond to error sources 63 to 32. 0: Error signal output not masked 1: Error signal output masked |

NOTE

If an error flag is set but masked, clearing the mask will set the $\overline{\text{ERROROUT}}/\overline{\text{ERROROUT_C}}$ to low level. This happens independently from the time of the error occurrence. In other words, the flag is evaluated statically.

32.4.20 ECMEMK2 — ECM Error Mask Register 2

The ECM error mask register 2 is used to mask the individual error sources of the error pin output(s).

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 30_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | ECMEMK229 | ECMEMK228 | ECMEMK227 | ECMEMK226 | ECMEMK225 | ECMEMK224 | ECMEMK223 | ECMEMK222 | ECMEMK221 | ECMEMK220 | ECMEMK219 | ECMEMK218 | ECMEMK217 | ECMEMK216 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMEMK215 | ECMEMK214 | ECMEMK213 | ECMEMK212 | ECMEMK211 | ECMEMK210 | ECMEMK209 | ECMEMK208 | ECMEMK207 | ECMEMK206 | ECMEMK205 | ECMEMK204 | ECMEMK203 | ECMEMK202 | ECMEMK201 | ECMEMK200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.34 ECMEMK2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------------|--|
| 31, 30 | Reserved | When writing, write the value after reset. |
| 29 | ECMEMK229 | Corresponds to delay timer overflow. 0: Error signal output is not masked. 1: Error signal output is masked. |
| 28 to 0 | ECMEMK228 to ECMEMK200 | ECM error output signal mask control bit ECMEMK228 to ECMEMK200 correspond to error sources 92 to 64. 0: Error signal output not masked 1: Error signal output masked |

NOTE

If an error flag is set but masked, clearing the mask will set the $\overline{\text{ERROROUT}}/\overline{\text{ERROROUT_C}}$ to low level. This happens independently from the time of the error occurrence. In other words, the flag is evaluated statically.

32.4.21 ECMESSTC0 — ECM Error Source Status Clear Trigger Register 0

The ECM error source status clear trigger register 0 is a write-only register and can be written in 32-bit units. This register is used to clear the individual error source status of the ECM master/checker error source status register 0. Both the error status of the ECM master and the ECM checker are cleared simultaneously.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 34_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | ECMCL SSE031 | ECMCL SSE030 | ECMCL SSE029 | ECMCL SSE028 | ECMCL SSE027 | ECMCL SSE026 | ECMCL SSE025 | ECMCL SSE024 | ECMCL SSE023 | ECMCL SSE022 | ECMCL SSE021 | ECMCL SSE020 | ECMCL SSE019 | ECMCL SSE018 | ECMCL SSE017 | ECMCL SSE016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMCL SSE015 | ECMCL SSE014 | ECMCL SSE013 | ECMCL SSE012 | ECMCL SSE011 | ECMCL SSE010 | ECMCL SSE009 | ECMCL SSE008 | ECMCL SSE007 | ECMCL SSE006 | ECMCL SSE005 | ECMCL SSE004 | ECMCL SSE003 | ECMCL SSE002 | ECMCL SSE001 | ECMCL SSE000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.35 ECMESSTC0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------------|---|
| 31 to 0 | ECMCLSSE031 to ECMCLSSE000 | ECM error status clear bit ECMCLSSE031 to ECMCLSSE000 correspond to ECMmSSE031 to ECMmSSE000. 0: Corresponding error status unchanged 1: Corresponding error status cleared |

32.4.22 ECMESSTC1 — ECM Error Source Status Clear Trigger Register 1

The ECM error source status clear trigger register 1 is a write-only register and can be written in 32-bit units. This register is used to clear the individual error source status of the ECM master/checker error source status register 1. Both the error status of the ECM master and the ECM checker are cleared simultaneously.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 38_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | ECMCL SSE131 | ECMCL SSE130 | ECMCL SSE129 | ECMCL SSE128 | ECMCL SSE127 | ECMCL SSE126 | ECMCL SSE125 | ECMCL SSE124 | ECMCL SSE123 | ECMCL SSE122 | ECMCL SSE121 | ECMCL SSE120 | ECMCL SSE119 | ECMCL SSE118 | ECMCL SSE117 | ECMCL SSE116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMCL SSE115 | ECMCL SSE114 | ECMCL SSE113 | ECMCL SSE112 | ECMCL SSE111 | ECMCL SSE110 | ECMCL SSE109 | ECMCL SSE108 | ECMCL SSE107 | ECMCL SSE106 | ECMCL SSE105 | ECMCL SSE104 | ECMCL SSE103 | ECMCL SSE102 | ECMCL SSE101 | ECMCL SSE100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.36 ECMESSTC1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------------|---|
| 31 to 0 | ECMCLSSE131 to ECMCLSSE100 | ECM error status clear bit ECMCLSSE131 to ECMCLSSE100 correspond to ECMmSSE131 to ECMmSSE100. 0: Corresponding error status unchanged 1: Corresponding error status cleared |

32.4.23 ECMESSTC2 — ECM Error Source Status Clear Trigger Register 2

The ECM error source status clear trigger register 2 is a write-only register and can be written in 32-bit units. This register is used to clear the individual error source status of the ECM master/checker error source status register 2. Both the error status of the ECM master and the ECM checker are cleared simultaneously.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 3C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | — | ECM CLSSE 230 | ECM CLSSE 229 | ECM CLSSE 228 | ECM CLSSE 227 | ECM CLSSE 226 | ECM CLSSE 225 | ECM CLSSE 224 | ECM CLSSE 223 | ECM CLSSE 222 | ECM CLSSE 221 | ECM CLSSE 220 | ECM CLSSE 219 | ECM CLSSE 218 | ECM CLSSE 217 | ECM CLSSE 216 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECM CLSSE 215 | ECM CLSSE 214 | ECM CLSSE 213 | ECM CLSSE 212 | ECM CLSSE 211 | ECM CLSSE 210 | ECM CLSSE 209 | ECM CLSSE 208 | ECM CLSSE 207 | ECM CLSSE 206 | ECM CLSSE 205 | ECM CLSSE 204 | ECM CLSSE 203 | ECM CLSSE 202 | ECM CLSSE 201 | ECM CLSSE 200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.37 ECMESSTC2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------------------|---|
| 31 | Reserved | When writing, write the value after reset. |
| 30 | ECMCLSSE230 | ECM error status clear bit ECMCLSSE230 corresponds to ECMmSSE230. 0: Error status unchanged 1: Error status cleared |
| 29 | ECMCLSSE229 | ECM error status clear bit ECMCLSSE229 corresponds to ECMmSSE229. 0: Error status unchanged 1: Error status cleared |
| 28 to 0 | ECMCLSSE228 to ECMCLSSE200 | ECM error status clear bit ECMCLSSE228 to ECMCLSSE200 correspond to ECMmSSE228 to ECMmSSE200. 0: Corresponding error status unchanged 1: Corresponding error status cleared |

32.4.24 ECMPCMD1 — ECM Protection Command Register

The ECM protection command register is a write-only register and can be written in 32-bit units. Refer to **Section 32.4.1, List of Registers**, for the protected registers. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence. The value after reset is undefined.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 40_H

Value after reset: Undefined

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|----|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | Undefined | | | | | | | | | | | | | | | |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | ECMREG1[7:0] | | | | | | | |
| Value after reset | Undefined | | | | | | | | | | | | | | | |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.38 ECMPCMD1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 8 | Reserved | When writing, write 0. |
| 7 to 0 | ECMREG1[7:0] | Protection command that enables writing to write protected ECM registers. |

32.4.25 ECMP5 — ECM Protection Status Register

The ECM protection status register is a read-only register. This register is used to verify whether the write protected register has been written successfully or not. Refer to **Section 32.3.5, Writing to Protected Registers**.

Access: This register can be read in 8-bit units.

Address: <ECM_base> + 44_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ECMPRERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

Table 32.39 ECMP5 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 7 to 1 | Reserved | When read, the value after reset is returned. |
| 0 | ECMPRERR | ECM protection status bit Indicates whether writing to a write protected register failed or succeeded. 0: Writing was successfully completed. 1: Writing failed |

32.4.26 ECMPE0 — ECM Pseudo Error Trigger Register 0

The ECM pseudo error trigger register 0 is a write-only register. This register is used to generate a pseudo error for test purposes. The ECM operation in response to the generation of a pseudo error is identical to that of a real error source.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 48_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | ECMPE031 | ECMPE030 | ECMPE029 | ECMPE028 | ECMPE027 | ECMPE026 | ECMPE025 | ECMPE024 | ECMPE023 | ECMPE022 | ECMPE021 | ECMPE020 | ECMPE019 | ECMPE018 | ECMPE017 | ECMPE016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMPE015 | ECMPE014 | ECMPE013 | ECMPE012 | ECMPE011 | ECMPE010 | ECMPE009 | ECMPE008 | ECMPE007 | ECMPE006 | ECMPE005 | ECMPE004 | ECMPE003 | ECMPE002 | ECMPE001 | ECMPE000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.40 ECMPE0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|---|
| 31 to 0 | ECMPE031 to ECMPE000 | ECM pseudo error trigger bit ECMPE031 to ECMPE000 correspond to error sources 31 to 0. 0: Pseudo error is not generated. 1: Pseudo error is generated. |

32.4.27 ECMPE1 — ECM Pseudo Error Trigger Register 1

The ECM pseudo error trigger register 1 is a write-only register. This register is used to generate a pseudo error for test purposes. The ECM operation in response to the generation of a pseudo error is identical to that of a real error source.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 4C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ECMPE 131 | ECMPE 130 | ECMPE 129 | ECMPE 128 | ECMPE 127 | ECMPE 126 | ECMPE 125 | ECMPE 124 | ECMPE 123 | ECMPE 122 | ECMPE 121 | ECMPE 120 | ECMPE 119 | ECMPE 118 | ECMPE 117 | ECMPE 116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMPE 115 | ECMPE 114 | ECMPE 113 | ECMPE 112 | ECMPE 111 | ECMPE 110 | ECMPE 109 | ECMPE 110 | ECMPE 107 | ECMPE 106 | ECMPE 105 | ECMPE 104 | ECMPE 103 | ECMPE 102 | ECMPE 101 | ECMPE 100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.41 ECMPE1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|--|
| 31 to 0 | ECMPE131 to ECMPE100 | ECM pseudo error trigger bit ECMPE131 to ECMPE100 correspond to error sources 63 to 32. 0: Pseudo error is not generated. 1: Pseudo error is generated. |

32.4.28 ECMPE2 — ECM Pseudo Error Trigger Register 2

The ECM pseudo error trigger register 2 is a write-only register. This register used to generate a pseudo error for test purposes. The ECM operation in response to the generation of a pseudo error is identical to that of a real error source.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 50_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | — | — | ECM PE229 | ECM PE228 | — | ECM PE226 | ECM PE225 | ECM PE224 | ECM PE223 | ECM PE222 | ECM PE221 | ECM PE220 | ECM PE219 | ECM PE218 | ECM PE217 | ECM PE216 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | W | W | R | W | W | W | W | W | W | W | W | W | W | W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECM PE215 | ECM PE214 | ECM PE213 | ECM PE212 | ECM PE211 | ECM PE210 | ECM PE209 | ECM PE208 | ECM PE207 | ECM PE206 | ECM PE205 | ECM PE204 | ECM PE203 | ECM PE202 | ECM PE201 | ECM PE200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

Table 32.42 ECMPE2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 31, 30 | Reserved | When writing, write the value after reset. |
| 29 | ECMPE229 | ECM pseudo error trigger bit. Corresponds to delay timer overflow. 0: Pseudo error is not generated. 1: Pseudo error is generated. |
| 28 | ECMPE228 | ECM pseudo error trigger bit. Corresponds to ECM compare error. 0: Pseudo error is not generated. 1: Pseudo error is generated. |
| 27 | Reserved | When writing, write the value after reset. |
| 26 to 0 | ECMPE226 to ECMPE200 | ECM pseudo error trigger bit ECMPE226 to ECMPE200 correspond to error sources 90 to 64. 0: Pseudo error is not generated. 1: Pseudo error is generated. |

NOTE

After Field-BIST executes, regardless of whether it is an intended execution or not, ECMmESSTR2.ECMmSSE227 will be always “1”.

In the startup (after reset release), Field-BIST will be always executed and “1” will be captured in ECMmESSTR2.ECMmSSE227. If ECMmESSTR2.ECMmSSE227 is not set to 1 after Field-BIST is run, then ECMmESSTR2.ECMmSSE227 has a failure. That is, it is possible to detect the failure of bit 27 of ECMmESSTR2 without conducting self-diagnosis using ECMPE2. For this reason, bit 27 of ECMPE2 is reserved.

32.4.29 ECMDTMCTL — ECM Delay Timer Control Register

The ECM delay timer control register is a read/write register. This register is used to control the delay timer.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 8-bit units.

Address: <ECM_base> + 54_H

Value after reset: 00_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|--------------|---|---|--------|--------|
| | — | — | — | DTMSTACNTCLK | — | — | ECMSTP | ECMSTA |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | W | R/W |

Table 32.43 ECMDTMCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 7 to 5 | Reserved | When writing, write the value after reset. |
| 4 | DTMSTACNTCLK | Delay timer start confirmation status. 0: Delay timer does not start. 1: Delay timer starts. |
| 1 | ECMSTP | Delay timer stop bit By writing “1” to this bit, the delay timer is initialized and stopped (writing 0 is ignored). Simultaneously, ECMSTA bit will be 0. 0: Delay timer is completed or stop request is not in progress. 1: Stop request for delay timer is in progress. |
| 0 | ECMSTA | Delay timer start bit Specifies the operation of the delay timer at the occurrence of an error event. 0: Delay timer does not start 1: Delay timer starts (Writing 0 to this bit stops the delay timer) |

NOTE

1. ECMDTMCTL register can be accessed via P-Bus but delay timer runs. ECMDTMCTL register is set via PBUS, however actual delay timer exists in a different clock domain. Therefore, time lag exists between writing of ECMDTMCTL and actual starting/stopping of delay timer.
DTMSTACNTCLK can be used to confirm whether delay timer is enabled or not.
Reconfirm that ECMSTA has been updated by checking DTMSTACNTCLK after writing to ECMSTA.
2. ECMDTMCTL register can be written only when (ECMSTA, DTMSTACNTCLK) = (0,0) or (1,1) Confirm the values of ECMSTA and DTMSTACNTCLK before writing to ECMDTMCTL.

32.4.30 ECMDTMR — ECM Delay Timer Register

The ECM delay timer register indicates the value of the delay timer counter. The ECM delay timer register is initialized by setting the ECMSTA bit of the ECM delay timer control register from 1 (timer in operation) to 0 (timer stopped). The ECM delay timer uses the internal oscillator clock divided by 2 for counting.

Access: This register can be read in 16-bit units.

Address: <ECM_base> + 58_H

Value after reset: 0000_H

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | ECMDTMR[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

32.4.31 ECMDTMCMP — ECM Delay Timer Compare Register

The ECM delay timer compare register is a read/write register. The ECMmSSE229 bit is set when the value in ECMDTMCMP register matches with the value of the ECM delay timer register. Writing data to this register has to be conducted while the delay timer is stopped.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 5C_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CMPW |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMDTMCMP[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.44 ECMDTMCMP Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 31 to 17 | Reserved | When writing, write the value after reset. |
| 16 | CMPW | Indicates whether reflecting the new ECMDTMCMP value across clock domains is in progress. 0: Not in progress 1: In progress. While the bit is set, further setting to the ECMDTMCMP is prohibited. When writing, write the value after reset. |
| 15 to 0 | ECMDTMCMP [15:0] | Delay timer compare value |

NOTE

1. ECMDTMCMP register is set via PBUS, however actual delay timer exists in a different clock domain. When ECMDTMCMP is configured, the value is copied across clock domains to reflect to the delay timer, which takes a certain time. CMPW indicates whether reflecting the new ECMDTMCMP value across clock domains is in progress.
2. While CMPW is 1, further setting to ECMDTMCMP is ignored. Please confirm CMPW = 0, before writing to ECMDTMCMP.

32.4.32 ECMDTMCFG0 — ECM Delay Timer Configuration Register 0

The ECM delay timer configuration register 0 is used to enable/disable the delay timer start caused by EI level interrupts in response to errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 60_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | ECMTE031 | ECMTE030 | ECMTE029 | ECMTE028 | ECMTE027 | ECMTE026 | ECMTE025 | ECMTE024 | ECMTE023 | ECMTE022 | ECMTE021 | ECMTE020 | ECMTE019 | ECMTE018 | ECMTE017 | ECMTE016 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMTE015 | ECMTE014 | ECMTE013 | ECMTE012 | ECMTE011 | ECMTE010 | ECMTE009 | ECMTE008 | ECMTE007 | ECMTE006 | ECMTE005 | ECMTE004 | ECMTE003 | ECMTE002 | ECMTE001 | ECMTE000 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.45 ECMDTMCFG0 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------|--|
| 31 to 0 | ECMTE031 to ECMTE000 | ECM delay timer start control bit ECMTE031 to ECMTE000 correspond to EI level interrupts generated by error sources 31 to 0. 0: Delay timer start disabled 1: Delay timer start enabled |

NOTE

Do not enable delay timer start for clock monitor error events (ECMTE015 to ECMTE008).

32.4.33 ECMDTMCFG1 — ECM Delay Timer Configuration Register 1

The ECM delay timer configuration register 1 is used to enable/disable the delay timer start caused by EI level interrupts in response to errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 64_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | ECMTE131 | ECMTE130 | ECMTE129 | ECMTE128 | ECMTE127 | ECMTE126 | ECMTE125 | ECMTE124 | ECMTE123 | ECMTE122 | ECMTE121 | ECMTE120 | ECMTE119 | ECMTE118 | ECMTE117 | ECMTE116 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMTE115 | ECMTE114 | ECMTE113 | ECMTE112 | ECMTE111 | ECMTE110 | ECMTE109 | ECMTE108 | ECMTE107 | ECMTE106 | ECMTE105 | ECMTE104 | ECMTE103 | ECMTE102 | ECMTE101 | ECMTE100 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.46 ECMDTMCFG1 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|---|
| 31 to 0 | ECMTE131 to ECMTE100 | ECM delay timer start control bit ECMTE131 to ECMTE100 correspond to EI level interrupts generated by error sources 63 to 32. 0: Delay timer start disabled 1: Delay timer start enabled |

32.4.34 ECMDTMCFG2 — ECM Delay Timer Configuration Register 2

The ECM delay timer configuration register 2 is used to enable/disable the delay timer start caused by EI level interrupts in response to errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 68_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | — | — | — | ECMTE 228 | ECMTE 227 | ECMTE 226 | ECMTE 225 | ECMTE 224 | ECMTE 223 | ECMTE 222 | ECMTE 221 | ECMTE 220 | ECMTE 219 | ECMTE 218 | ECMTE 217 | ECMTE 216 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMTE 215 | ECMTE 214 | ECMTE 213 | ECMTE 212 | ECMTE 211 | ECMTE 210 | ECMTE 209 | ECMTE 208 | ECMTE 207 | ECMTE 206 | ECMTE 205 | ECMTE 204 | ECMTE 203 | ECMTE 202 | ECMTE 201 | ECMTE 200 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.47 ECMDTMCFG2 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|---|
| 31 to 29 | Reserved | When writing, write the value after reset. |
| 28 to 0 | ECMTE228 to ECMTE200 | ECM delay timer start control bit ECMTE228 to ECMTE200 correspond to EI level interrupts generated by error sources 92 to 64. 0: Delay timer start disabled 1: Delay timer start enabled |

32.4.35 ECMDTMCFG3 — ECM Delay Timer Configuration Register 3

The ECM delay timer configuration register 3 is a read/write register and can be written in 32-bit units. This register is used to enable/disable the delay timer start caused by Non-maskable interrupt in response to errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 6C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ECMTE 331 | ECMTE 330 | ECMTE 329 | ECMTE 328 | ECMTE 327 | ECMTE 326 | ECMTE 325 | ECMTE 324 | ECMTE 323 | ECMTE 322 | ECMTE 321 | ECMTE 320 | ECMTE 319 | ECMTE 318 | ECMTE 317 | ECMTE 316 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMTE 315 | ECMTE 314 | ECMTE 313 | ECMTE 312 | ECMTE 311 | ECMTE 310 | ECMTE 309 | ECMTE 308 | ECMTE 307 | ECMTE 306 | ECMTE 305 | ECMTE 304 | ECMTE 303 | ECMTE 302 | ECMTE 301 | ECMTE 300 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.48 ECMDTMCFG3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|--|
| 31 to 0 | ECMTE331 to ECMTE300 | ECM delay timer start control bit ECMTE331 to ECMTE300 correspond to FE level interrupts generated by error sources 31 to 0. 0: Delay timer start disabled 1: Delay timer start enabled |

NOTE

Do not enable delay timer start for clock monitor error events (ECMTE315 to ECMTE308).

32.4.36 ECMDTMCFG4 — ECM Delay Timer Configuration Register 4

The ECM delay timer configuration register 4 is a read/write register and can be written in 32-bit units. This register is used to enable/disable the delay timer start caused by Non-maskable interrupt in response to errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 70_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ECMTE 431 | ECMTE 430 | ECMTE 429 | ECMTE 428 | ECMTE 427 | ECMTE 426 | ECMTE 425 | ECMTE 424 | ECMTE 423 | ECMTE 422 | ECMTE 421 | ECMTE 420 | ECMTE 419 | ECMTE 418 | ECMTE 417 | ECMTE 416 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ECMTE 415 | ECMTE 414 | ECMTE 413 | ECMTE 412 | ECMTE 411 | ECMTE 410 | ECMTE 409 | ECMTE 408 | ECMTE 407 | ECMTE 406 | ECMTE 405 | ECMTE 404 | ECMTE 403 | ECMTE 402 | ECMTE 401 | ECMTE 400 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.49 ECMDTMCFG4 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------------|---|
| 31 to 0 | ECMTE431 to ECMTE400 | ECM delay timer start control bit ECMTE431 to ECMTE400 correspond to non-maskable interrupts generated by error sources 63 to 32. 0: Delay timer start disabled 1: Delay timer start enabled |

32.4.37 ECMDTMCFG5 — ECM Delay Timer Configuration Register 5

The ECM delay timer configuration register 5 is a read/write register and can be written in 32-bit units. This register is used to enable/disable the delay timer start caused by Non-maskable interrupt in response to errors.

You have to follow Protection Unlock Sequence for writing data to this register. Refer to **Section 32.3.5, Writing to Protected Registers** for details of the Protection Unlock Sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 74_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | — | — | — | ECMTE528 | ECMTE527 | ECMTE526 | ECMTE525 | ECMTE524 | ECMTE523 | ECMTE522 | ECMTE521 | ECMTE520 | ECMTE519 | ECMTE518 | ECMTE517 | ECMTE516 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMTE515 | ECMTE514 | ECMTE513 | ECMTE512 | ECMTE511 | ECMTE510 | ECMTE509 | ECMTE508 | ECMTE507 | ECMTE506 | ECMTE505 | ECMTE504 | ECMTE503 | ECMTE502 | ECMTE501 | ECMTE500 |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.50 ECMDTMCFG5 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------------|---|
| 31 to 29 | Reserved | When writing, write the value after reset. |
| 28 to 0 | ECMTE528 to ECMTE500 | ECM delay timer start control bit ECMTE528 to ECMTE500 correspond to non-maskable interrupts generated by error sources 92 to 64. 0: Delay timer start disabled 1: Delay timer start enabled |

32.4.38 ECMEOCCFG — ECM Error Output Clear Invalidation Configuration Register

This register is a read/write register and can be written in 32-bit units.

The register can determine the period to suppress clearing $\overline{\text{ERROROUT}}/\overline{\text{ERROROUR_C}}$ state after an error occurs.

When the value of the error output clear invalidation counter exceeds the value configured in this register, the error status of the $\overline{\text{ERROROUT}}/\overline{\text{ERROROUR_C}}$ pin can be cleared by software.

Configure this register only when $\overline{\text{ERROROUT}}/\overline{\text{ERROROUR_C}}$ pin status is not in error status.

This register is initialized by Power on Reset or System Reset 1 or System Reset 2.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 78_H

Value after reset: 0000 0000_H This register is initialized by Power on Reset or System Reset 1 or System Reset 2.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CMPW |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ECMEOUTCLRT[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 32.51 ECMEOCCFG Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------------|---|
| 31 to 17 | Reserved | When writing, write the value after reset. |
| 16 | CMPW | Indicates whether reflecting the new ECMEOUTCLRT across clock domains is in progress 0: Not in progress 1: In progress. While the bit is set, further setting to the ECMEOUTCLRT is prohibited. When writing, write the value after reset. |
| 15 to 0 | ECMEOUTCLRT[15:0] | The number of clock cycles after which it is possible to clear error output by software. |

NOTES

1. ECMEOUTCLRT register is set via PBUS, however actual “output clear invalidation counter” exists in a different clock domain. When ECMEOUTCLRT is configured, the value is copied across clock domains to reflect to the counter, which takes a certain time. CMPW indicates whether the reflection across clock domains is in progress.
2. While CMPW is 1, further setting to ECMEOUTCLRT is ignored. Please confirm CMPW = 0, before writing to ECMEOUTCLRT.

32.4.39 ECMPEM — ECM Pseudo Error Mask Register

This register can mask the pseudo error of “ECM compare error” to support self-diagnosis of the binding components for ERROROUT and ERROROUR_C pins.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 7C_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSKM | MSKC |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

Table 32.52 ECMPEM Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 2 | Reserved | When writing, write the value after reset. |
| 1 | MSKM | 0: Pseudo error of “ECM compare error” for ECM master is NOT masked. 1: Pseudo error of “ECM compare error” for ECM master is masked. |
| 0 | MSKC | 0: Pseudo error of “ECM compare error” for ECM checker is NOT masked. 1: Pseudo error of “ECM compare error” for ECM checker is masked. |

32.4.40 ECMEPCTL — ECM Error Pulse Control Register

This register selects the timer output in dynamic mode.

Access: This register can be read/written in 8-bit units.

Address: FFD6 3000_H

Value after reset: 00_H

| | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | ECMTMSL |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W |

Table 32.53 ECMEPCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 7 to 1 | Reserved | When writing, write the value after reset. |
| 0 | ECMTMSL | Timer output selection to ERROROUT/ERROROUT_C 0: Channel 15 in TAUD1 1: OSTM1 |

Section 33 Data CRC Function A (DCRA)

Data CRC function (DCRA) can be used to verify or generate CRC (cyclic redundancy check) protecting a variety of data blocks.

After initialization of the DCRA data register, every write access to the DCRA input register generates a new CRC according to the chosen polynomial and the result is stored in the DCRA data register.

The CRC for any data of arbitrary bytes can be calculated with selectable CRC generator polynomials. DCRA is implemented 4 units.

33.1 Features of RH850/P1M-E DCRA

33.1.1 Units and Channels

This microcontroller has the following number of DCRA units.

Table 33.1 Number of Units

| Product | RH850/P1M-E 100 pins | RH850/P1M-E 144 pins |
|-----------------|-------------------------|-------------------------|
| Number of Units | 4 | 4 |
| Name | DCRAn (n = 0 to 3) | DCRAn (n = 0 to 3) |

Table 33.2 Unit Configurations and Channels

| Unit Name (Channel Name) DCRAn | Channels per Unit | RH850/P1M-E 100 pins (4 ch) | RH850/P1M-E 144 pins (4 ch) |
|--------------------------------------|----------------------|-----------------------------------|-----------------------------------|
| DCRA0 | 1 | √ | √ |
| DCRA1 | 1 | √ | √ |
| DCRA2 | 1 | √ | √ |
| DCRA3 | 1 | √ | √ |

Table 33.3 Index

| Index | Description |
|-------|--|
| n | Throughout this section, the individual Data CRC function A units are identified by the index "n" (n = 0 to 3); for example, DCRAnCTL indicates the CRCn control register. |

33.1.2 Register Base Address

DCRAn base addresses are listed in the following table.

DCRAn register addresses are given as offsets from the base addresses throughout this section.

Table 33.4 Register Base Addresses

| Base Address Name | Base Address |
|-------------------|------------------------|
| <DCRA0_base> | FFD5 0000 _H |
| <DCRA1_base> | FFD5 1000 _H |
| <DCRA2_base> | FFD5 2000 _H |
| <DCRA3_base> | FFD5 3000 _H |

33.1.3 Clock Supply

Clock supply by and to DCRAn is listed in the following table.

Table 33.5 Clock Supply

| Unit Name | Unit Clock Name | Supply Clock Name |
|-----------|-----------------|-------------------------------------|
| DCRAn | PCLK | High-speed peripheral clock CLK_HSB |

33.1.4 Interrupt Requests

DCRAn has no interrupt request.

33.1.5 Reset Sources

Please refer to **Section 8, Reset Controller**.

33.1.6 External Input/Output Signals

DCRAn has no external pin.

33.2 Overview

33.2.1 Functional Overview

The Data CRC function A can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- 32-bit Ethernet CRC
($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1$)
- 16-bit CCITT CRC
($X^{16} + X^{12} + X^5 + 1$)
- 8-bit SAE J1850
($X^8 + X^4 + X^3 + X^2 + 1$)
- 8-bit 0x2F
($X^8 + X^5 + X^3 + X^2 + X^1 + 1$)
- CRC generation to an arbitrary data block length.
- After initialization of the CRC input register every write access to the CRC data register generates a new CRC according to the chosen polynomial and the result is stored in the CRC data register.

33.2.2 Block Diagram

The following picture shows the block diagram of the Data CRC function A.

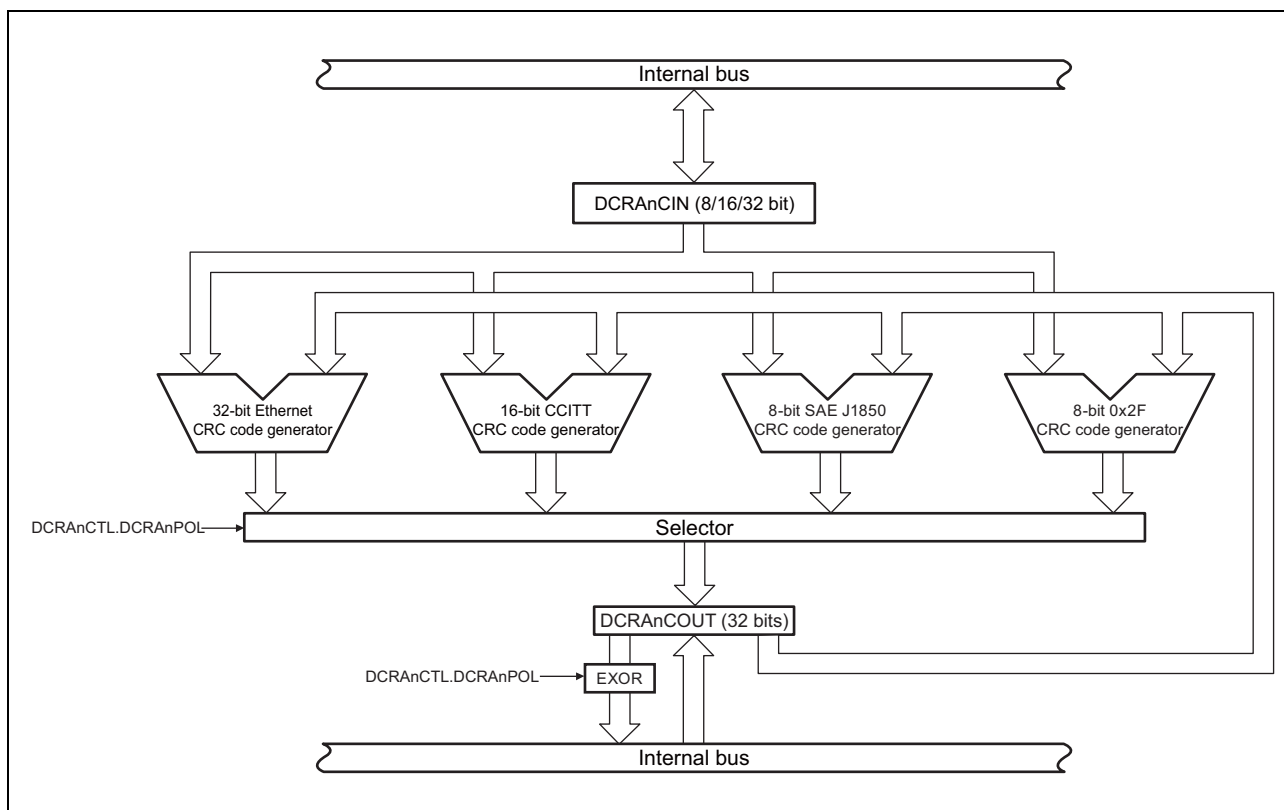
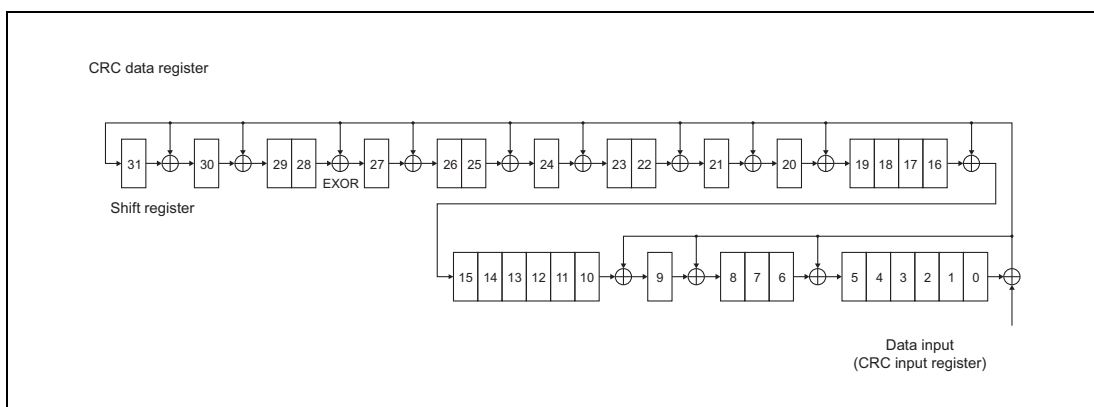


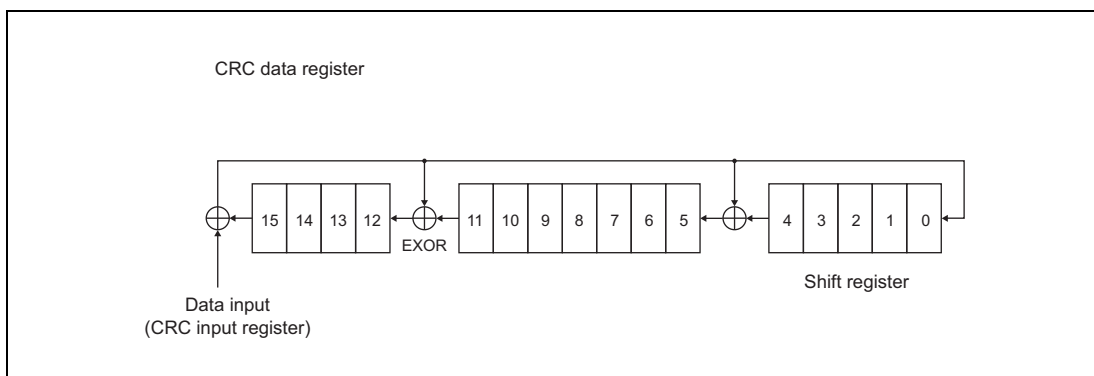
Figure 33.1 Block Diagram of Data CRC Function A

33.2.3 Operation Circuit

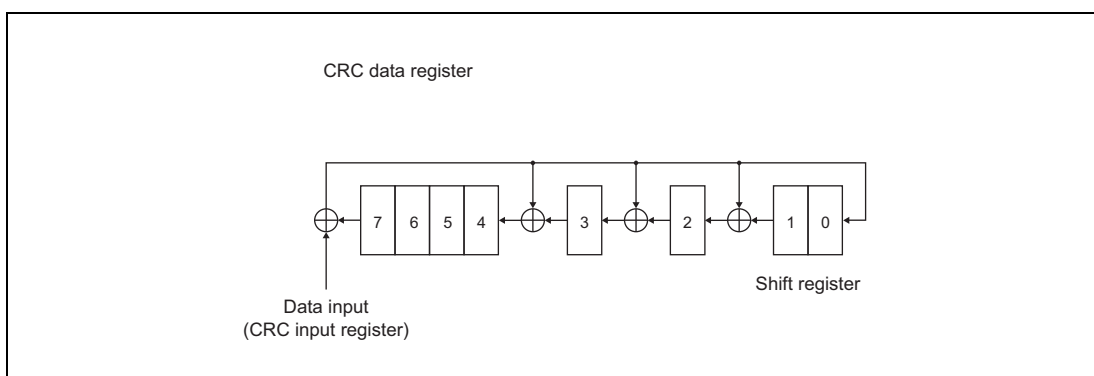
- 32-bit ethernet



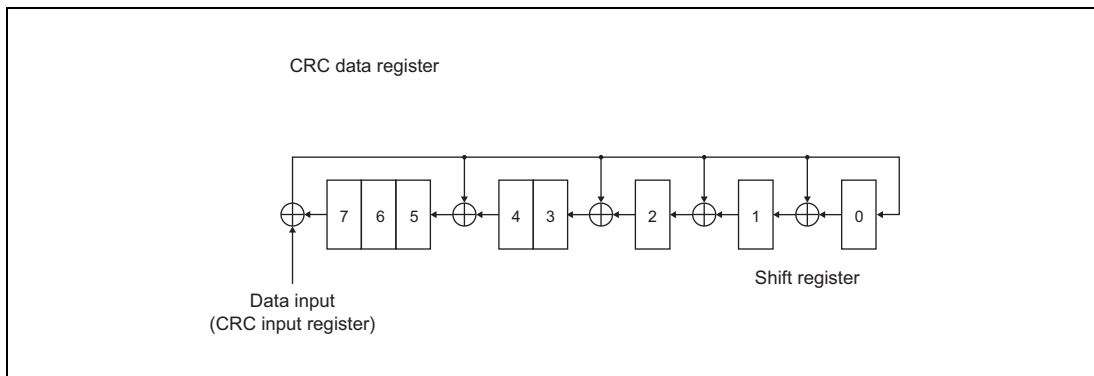
- 16-bit CCITT



- 8-bit SAE J1850



- 8-bit 0x2F



33.3 Registers

33.3.1 List of Registers

DCRA registers are listed in the following table.

<DCRAn_base> is defined in **Section 33.1.2, Register Base Address**.

Table 33.6 Registers

| Module | Register | Symbol | Address |
|--------|-----------------------|-----------|--------------------------------|
| DCRAn | CRCn input register | DCRAnCIN | <DCRAn_base> + 00 _H |
| DCRAn | CRCn data register | DCRAnCOUT | <DCRAn_base> + 04 _H |
| DCRAn | CRCn control register | DCRAnCTL | <DCRAn_base> + 20 _H |

33.3.2 DCRAnCIN — CRCn Input Register

This register holds the input data for the CRC calculation. The effective bit width used for CRC calculation must be set by DCRAnCTL.DCRAnISZ[1:0].

When data is written to this register, the CRC code is generated.

The CRC calculation is immediately started after the DCRAnCIN register is written. The DCRAnCOUT register must be initialized, with the initial starting value, before the first data of the data block is written to DCRAnCIN register.

Access: This register can be read/written in 32-bit units.

Address: <DCRAn_base>

Value after reset: 0000 0000_H This register is initialized by any reset.

| | | | | | | | | | | | | | | | | |
|-------------------|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | DCRAnCIN[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DCRAnCIN[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Table 33.7 DCRAnCIN Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------------|--|
| 31 to 0 | DCRAnCIN [31:0] | Input Data for CRC Calculation The valid bits are: <ul style="list-style-type: none"> • For 32 bit effective bit width: DCRAnCIN[31:0] • For 16 bit effective bit width: DCRAnCIN[15:0] • For 8 bit effective bit width: DCRAnCIN[7:0] |

33.3.3 DCRAnCOUT — CRCn Data Register

This register stores the result of the CRC code generated by any of the following polynomials.

- 32-bit Ethernet
- 16-bit CCITT
- 8-bit SAE J1850
- 8-bit 0x2F

Access: This register can be read/written in 32-bit units.

Address: <DCRAn_base> + 4_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | DCRAnCOUT[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | DCRAnCOUT[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Note 1. The read value after reset is 0000 0000_H since the 32-bit ethernet CRC polynomial is selected as the CRC generating function after reset.

Table 33.8 DCRAnCOUT Register Contents

| Bit Position | Bit Name | Function |
|--------------|------------------|--|
| 31 to 0 | DCRAnCOUT [31:0] | <p>Result of the CRC Code Generation</p> <p>When the 16-bit CCITT polynomial is enabled, bits 15 to 0 show the CRC result. The values of bits 31 to 16 are undefined.</p> <p>When the 8-bit SAE J1850 or 0x2F polynomial is enabled, bits 7 to 0 show the CRC result. The values of bits 31 to 8 are undefined.</p> <p>The read value of this register is a value obtained by performing EXOR calculation for the following value:</p> <ul style="list-style-type: none"> • For 32-bit ethernet polynomial: FFFF FFFF_H • For 16-bit CCITT polynomial: 0000_H • For 8-bit SAE J1850 or 0x2F polynomial: 0000 00FF_H <p>For example, when DCRAnCOUT = 5555 5555_H for the 32-bit Ethernet polynomial, AAAA AAAA_H is read.</p> |

CAUTION

This register must be initialized by setting the initial start value before the first data of the data block is written to DCRAnCIN register.

33.3.4 DCRAnCTL — CRCn Control Register

This register controls the CRC generation process.

Access: This register can be read/written in 8-bit units.

Address: <DCRAn_base> + 20_H

Value after reset: 00_H This register is initialized by any reset.

| | | | | | | | | |
|-------------------|---|---|---------------|-----|---|---|---------------|-----|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | DCRAnISZ[1:0] | | — | — | DCRAnPOL[1:0] | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R/W | R/W |

Table 33.9 DCRAnCTL Register Contents

| Bit Position | Bit Name | Function |
|--------------|---------------|--|
| 7, 6 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 5, 4 | DCRAnISZ[1:0] | Specifies the CRC input bit width: 00: 32 bit (DCRAnCIN[31:0]) 01: 16 bit (DCRAnCIN[15:0]) 10: 8 bit (DCRAnCIN[7:0]) 11: Setting prohibited |
| 3, 2 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 1, 0 | DCRAnPOL[1:0] | Specifies the CRC generating function: 00: 32-bit ethernet CRC polynomial The byte order of the DCRAnCIN register is LSB (least significant byte) first. This means that if the CRC input bit width is 8 bits (DCRAnISZ[1:0] = 10 _B), the input values are in bits 7 to 0 of the DCRAnCIN register and bit 0 (the LSB) is the first bit of the input data. 01: 16-bit CCITT CRC polynomial The byte order of the DCRAnCIN register is MSB (most significant byte) first, meaning input values are in bits 7 to 0 of the DCRAnCIN register and bit 7 (the MSB) is the first bit of the input data if the CRC input bit width is 8 bits (DCRAnISZ[1:0] = 10 _B). 10: 8-bit SAE J1850 polynomial The byte order of the DCRAnCIN register is MSB (most significant byte) first, meaning input values are in bits 7 to 0 of the DCRAnCIN register and bit 7 (the MSB) is the first bit of the input data if the CRC input bit width is 8 bits (DCRAnISZ[1:0] = 10 _B). 11: 8-bit 0x2F polynomial The byte order of the DCRAnCIN register is MSB (most significant byte) first, meaning input values are in bits 7 to 0 of the DCRAnCIN register and bit 7 (the MSB) is the first bit of the input data if the CRC input bit width is 8 bits (DCRAnISZ[1:0] = 10 _B). |

NOTE

If the CRC generation method (DCRAnCTL.DCRAnPOL) is changed, the DCRAnCOUT register must be initialized by setting the initial start value.

CAUTION

The CRC input bit width (DCRAnCTL.DCRAnISZ[1:0]) must be set according to the data block bit width. Changing the CRC input bit width is not allowed during processing of a data block (a data block consists of N bytes, half-words or one word). After the final CRC result is read from DCRAnCOUT register, the CRC input bit width can be changed. In that case, the DCRAnCOUT register must be initialized with the initial start value.

33.4 Operation

The Data CRC Function A generates a CRC (cyclic redundancy check) value of an arbitrary block length. The data for the CRC function is forwarded in 8-, 16-, or 32-bit units. The CRC polynomial can either be selected from 32-bit ethernet CRC or 16-bit CCITT CRC. The initial starting value must be set at the DCRA_nCOUT register before the first write access to the CRC input register (DCRA_nCIN) is performed.

The flow chart below shows the CRC value generating procedure.

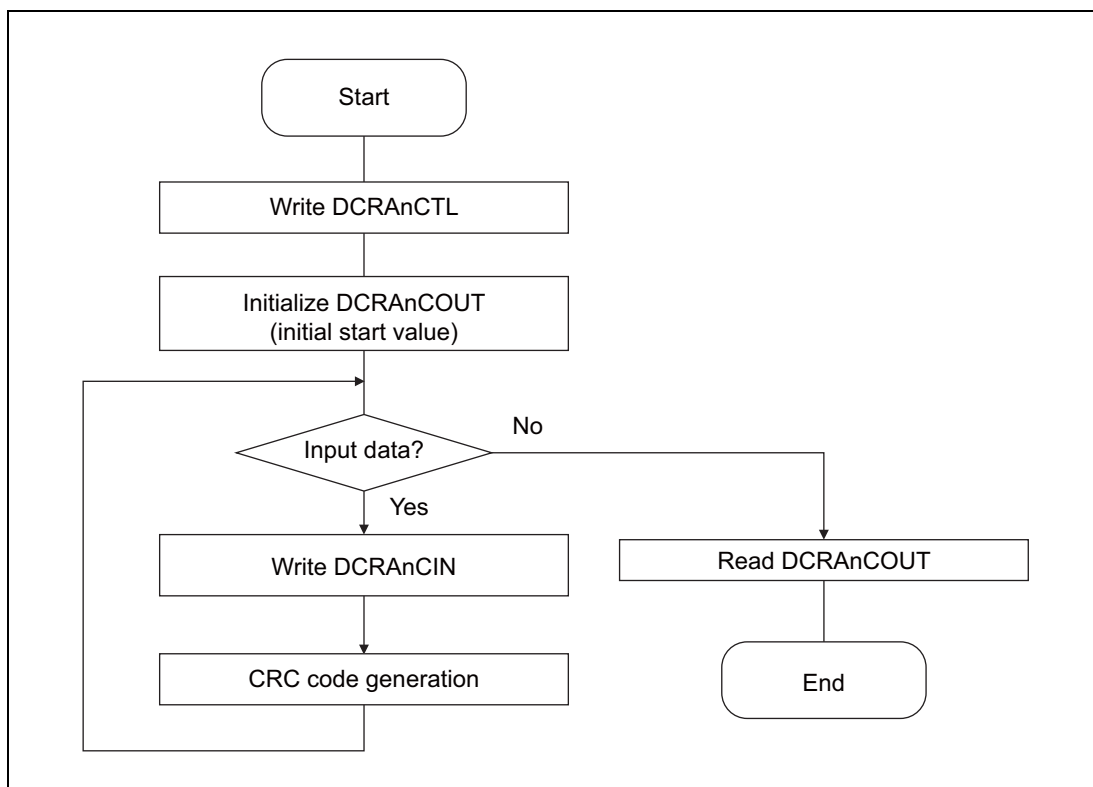


Figure 33.2 Data CRC Function A Flow Diagram

NOTES

1. Before writing the first data to DCRA_nCIN, the CRC output register DCRA_nCOUT must be initialized with the initial start value.
2. DCRA_nCOUT must be initialized by setting the initial start value when the polynomial is changed by changing DCRA_nCTL.DCRA_nPOL.
3. Setting example of the initial start values of the respective polynomials
The following is the example of setting values.

Table 33.10 Setting Example of the Initial Start Values (Reset when read)

| | Initial Start Value | EXOR Value | DCRA _n COUT Read Value |
|----------------------|------------------------|------------------------|-----------------------------------|
| 8-bit SAE J1850/0x2F | XXXX XXFF _H | XXXX XXFF _H | XXXX XX00 _H |
| 16-bit CCITT | XXXX FFFF _H | XXXX 0000 _H | XXXX FFFF _H |
| 32-bit ethernet | FFFF FFFF _H | FFFF FFFF _H | 0000 0000 _H |

Note: x: Undefined

Section 34 On-Chip Debugging Unit (OCD)

34.1 Debug Function

This product has an on-chip debug function. Using an on-chip debug emulator, the microcontroller included in the target system can debug its programs.

CAUTION

The debug function described in this section is supported by the microcontroller, but whether it can be used or not depends on the debugger. For details of the debugger, see the debugger's user's manual. For the necessary tool connection circuitry, please refer to the corresponding tool documentation.

(1) Debug Interface

This product supports the Nexus JTAG Interface, and Low Pin Debug Interface (4 pins) (hereinafter referred to as LPD (4 pins)) as the debug interfaces.

(2) Debug Monitor Function

In debug mode, the monitor program is run in the area dedicated to debugging.

Execution of the monitor program enables the following basic debug functions:

- Download of user programs
- Reading and writing of user resources including memory and registers while a user program is suspended.
- Execution of user programs starting from any addresses

(3) On-chip Break Function

Twelve break points are included in the CPU. Four of them can be designated for any accesses (access address and access data).

(4) Software Break Function

A software break point can be designated for any addresses on the user programs stored in the RAM.

(5) Forced Break Function

Execution of a user program can be forcibly suspended.

(6) Debug Interrupt Interface Function

Execution of a user program can be forcibly suspended by asserting an input to the $\overline{\text{EVTI}}$ pin from the outside.

(7) Forced Reset Function

The microcontroller (this product) can be forcibly reset by the Debugger Initiated Reset.

(8) Real-time RAM Monitor (RRM)

A memory can be read during execution of a program. The impact on program execution is minimal since this read access use a dedicated DMA unit for debugging.

(9) Dynamic Memory Modify (DMM)

A memory can be written during execution of a program. The impact on program execution is minimal since this write access use a dedicated DMA unit for debugging.

(10) Timer Function

Using a 32-bit counter, the execution time of a user program can be measured based on a dedicated debugging clock.

(11) Mask Function

Reset factors (pin reset, software reset, or ECM reset) can be masked.

(12) Event Detection Function

Events can be detected by the following: execution address, access address, access data, range (comparison in size), and sequential execution.

(13) Trigger Input Interface

This product incorporates an event trigger input interface to accept external events. It can accept an external event in response to an input to the $\overline{\text{EVTI}}$ pin.

(14) Trigger Output Interface

This product incorporates an event trigger output interface to notify the external debugger of event detection. Output from the $\overline{\text{EVTO}}$ pin can output a trigger of event detection to the outside.

(15) Hot Plug-in Function

Debugging can be started in normal operating mode without an input of a pin reset.

(16) Security Function

To prevent the contents of the flash memory from being read by an unauthorized person, a 128-bit ID code (OCD_ID) can be written to the microcontroller. If the code the user inputs when starting a debugger does not match the ID code, the flash memory cannot be accessed.

(17) Calibration Function

Emulation of the flash memory and tuning of ROM data can be executed by using the ERAM, the memory for emulation. For details, see **Section 34.2, Calibration Function**.

(18) Tracing Function

Information such as the user program execution history and variations in data can be obtained. For details, see **Section 34.3, Trace Control Function**.

34.2 Calibration Function

This product includes the Emulation RAM as an emulation memory for the on-chip flash memory.

(1) Emulation RAM

Products with 1-MB flash memory include 8-KB of emulation RAM with and products with 2-MB memory include 32-KB of emulation RAM.

(2) Flash Emulation Function

The emulation RAM can be mapped to any areas in the flash area. For details, see **Section 36.4, Emulation RAM**.

(3) Tuning Function

The ROM data can be dynamically tuned during execution of a user program via the Emulation RAM that has been mapped to the flash area.

(4) Flash Cache Clear Function

The flash cache is cleared when the Emulation RAM mapping is set. This preserves coherency of the contents between the on-chip flash memory or the Emulation RAM and the flash cache memory.

34.3 Trace Control Function

This product provides several trace functions including branch PC trace and data trace of the CPU and DMA data trace.

(1) Trace RAM

This product has 32 KB for the trace RAM.

The trace information in the trace RAM is accessible via the debug interfaces: Nexus, and LPD (4 pins).

(2) Software Trace

This function enables obtaining information such as the user program execution history and variations in data.

The software trace information can be output via the LPD debug interface.

34.4 Peripheral Break Control

Peripheral break is a function for stopping the peripheral modules when a user program is halted (at a break point, etc.).

The on-chip modules can be classified into two groups according to their operation in response to a peripheral break as follows:

1. Modules that are unconditionally stopped: WDTA0
2. Modules for which halting or continuing can be selected: OSTM0, OSTM1, OSTM3, OSTM4, OSTM5, OSTM6, OSTM7, TAPA0, TAPA1, TAPA2, TAPA3, TPBA0, TPBA1, TSG30, TSG31, ENCA0, ENCA1, TAUJ0, TAUJ1, TAUJ2, TAUD0, TAUD1, TAUD2, CSIG0, CSIH0, CSIH1, CSIH2, CSIH3, RLIN30, RLIN31

34.5 Usage Notes for On-chip Debugging

- (1) **Caution on Devices Used for Debugging**
Do not mount a device used for debugging on mass-produced products. The number of times data can be written to the flash memory cannot be guaranteed because the memory has already been rewritten during debugging.
- (2) When using a debugger, be aware that the execution will be started from the reset vector before preparation for communication between the OCD emulator and the microcomputer completes.

Section 35 Flash Memory

This section describes the flash memory mounted on RH850/P1M-E.

The first part in this section describes the characteristics of the mounted flash memory and the characteristics specific to RH850/P1M-E, such as the memory map, flash memory programming, and ECC.

35.1 Features

- Includes code flash memory and data flash memory
The code flash memory can store program codes and data and has the user area and the extended user area.
The data flash memory is used for storing data.
- Method of flash memory programming
Flash memory programming via a serial interface and programming of flash memory by a user program (self-programming) are supported.
- Support for BGO (Back Ground Operation)
The BGO function allows programs to be executed in the code flash memory while the data flash memory is being programmed/erased.
- Flash memory data security
 - Support for security functions to protect against illicit tampering with or reading out of data in the flash memory
 - Support for protection functions to protect against erroneous overwriting of the flash memory
- Option byte function
Sets the operation after releasing reset for ports, and WDTA, and external bus clocks.
- Support for the error detection/correction function (ECC) in the code flash memory and data flash memory
Built-in ECC function can detect 2-bit errors and detect/correct 1-bit errors.
- Interrupts can be acknowledged in self-programming mode.

For code flash sizes and data flash sizes of each product, see the following sections.

- **Section 4.1, Address Space**

35.2 Structure of Memory

35.2.1 Mapping of Code Flash Memory

Figure 35.1 illustrates the mapping of the code flash memory for the 2-Mbytes/1-Mbytes device. The user area of the code flash memory of the RH850/P1M-E is divided into 8-and 32-Kbyte blocks, which serve as the units of erasure. A single block of 32 Kbyte extended user area is also incorporated. The user area and extended user area are available as areas for storing the user program.

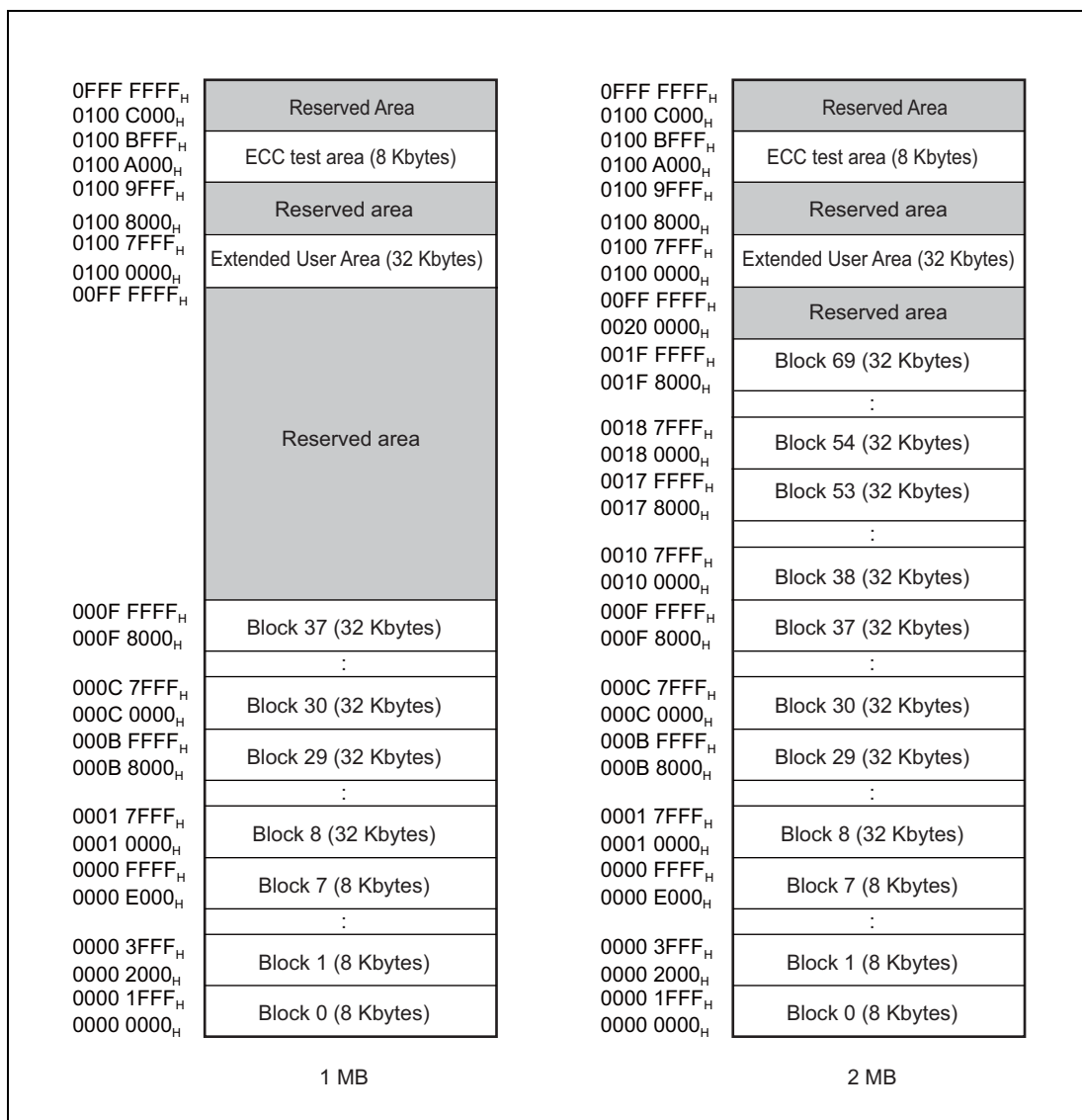


Figure 35.1 Mapping of the Code Flash Memory

35.2.2 Mapping of Data Flash Memory

The data area of the data flash memory in the RH850/P1M-E is divided into 64-byte blocks, with each being a unit for erasure. **Figure 35.2** shows the mapping of the data flash memory.

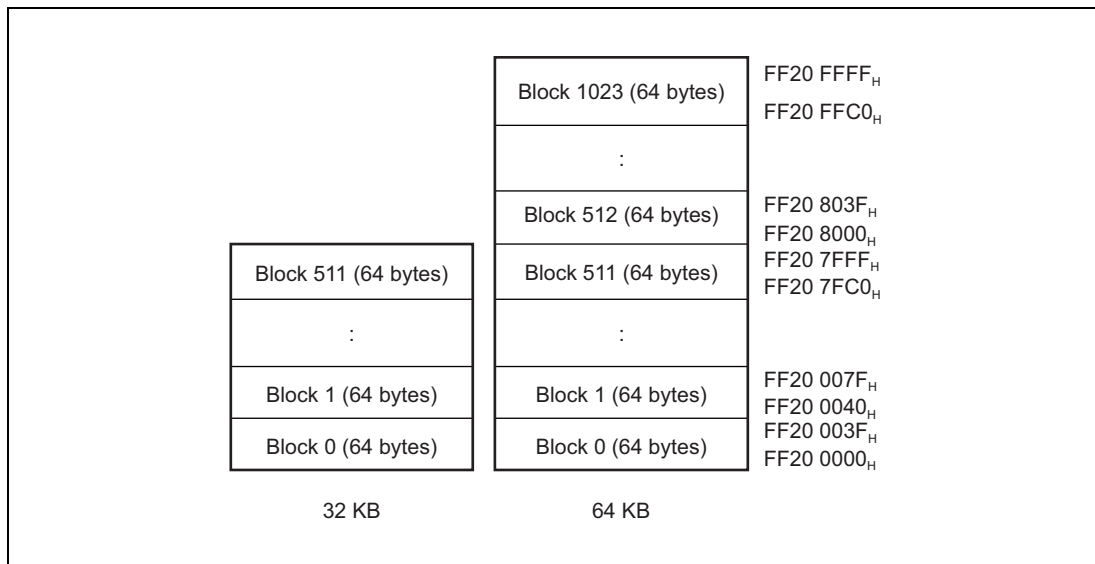


Figure 35.2 Mapping of the Data Flash Memory

35.3 Operating Modes Associated with Flash Memory

Figure 35.3 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see **Section 5, Operating Modes**.

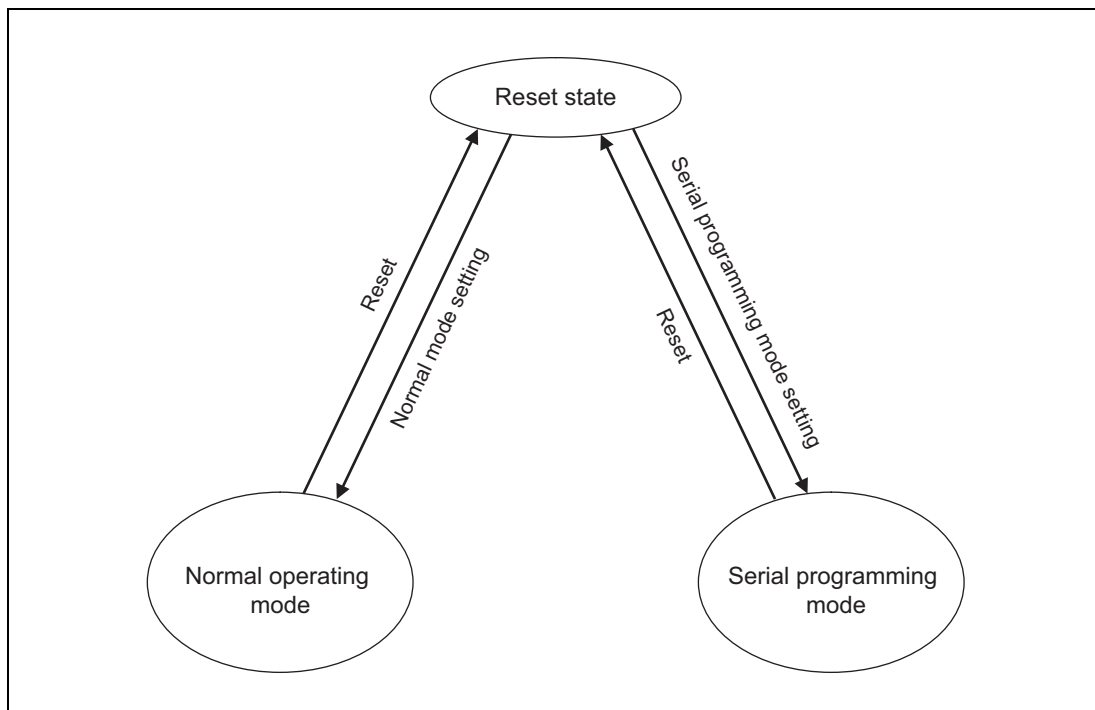


Figure 35.3 Mode Transition Associated with Flash Memory

Table 35.1 shows the flash memory area which is programmable and erasable in each mode and the boot program after reset release.

Table 35.1 Programmable and Erasable Area in Each Mode and the Boot Program after Reset Release

| Item | Normal Operating Mode | Serial Programming Mode |
|----------------------------------|--|--|
| Programmable and erasable area | <ul style="list-style-type: none"> • User area • Extended user area • Data area | <ul style="list-style-type: none"> • User area • Extended user area • Data area |
| Boot program after reset release | Program in user area or extended user area (Changeable by using the variable reset vector) | Firmware program for serial programming |

35.4 Functions

35.4.1 Functional Overview

On-chip flash memory of RH850/P1M-E can be programmed regardless of before and after the mounting on the target system with dedicated flash memory programmer via serial interface (serial programming).

Furthermore, security functions to prohibit updating of the user program written in the flash memory are incorporated, and this can prevent tampering by third parties.

Programming by the user program (self-programming) is suited for applications where the target system program may require updating after deployed to the end user. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as external communications, etc., and this allows programming under various conditions.

Table 35.2 gives an overview of the methods of programming and the corresponding operating modes.

Table 35.2 Methods of Programming

| Method of Programming | Overview of Functionality | Operating Mode |
|-----------------------|---|-------------------------|
| Serial programming | A dedicated flash memory programmer allows on-board programming of the flash memory after the device is mounted on the target system. | Serial programming mode |
| | A dedicated flash memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system. | |
| Self-programming | The user program that is written to code flash memory in advance by serial programming also allows updating the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data in code flash memory while the data flash memory is self-programming. For this reason, it is possible to update the data flash memory by executing a program written to the code flash memory. Instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being updated by self-programming. In such cases, a program for updating must be transferred to the local RAM in advance and executed. | Normal operating mode |

Table 35.3 lists the functions of the flash memory. Dedicated flash memory programmer commands enable serial programming, while reading of the flash memory by the user program enables self-programming.

Table 35.3 Basic Functions at a Glance

| Function | Description in Overview | Level of Support (√: Supported, Δ: Conditionally Supported, —: Not Supported) | |
|--|--|---|--|
| | | Serial programming | Self-programming |
| Blank checking | This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from code flash memory and data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure. | √ | √ |
| Block erasure | This is for erasing the contents of a specified block of memory. | √ | √ |
| Programming | This is for writing to a specified address. | √ | √ |
| Verification and checksum | Data that are read out from flash memory are compared with data transferred from the flash memory programmer. | √ | — (Reading of data by the user program is possible) |
| Reading | Data that have been written to the flash memory are read out. | √ | √ |
| Setting for OTP (one-time programming) | A specified block of code flash memory is set for OTP (OTP can only be set, that is, it is not possible to release a block's OTP setting). | √ | √ |
| Setting an ID | An ID setting is made for use in controlling the connection of a dedicated flash memory programmer for serial programming, controlling of the on-chip debugger, and programming of the code flash memory by self-programming. | √ | √ |
| Security settings | Security settings are for use in serial programming. | √ | Δ (Only when setting is prohibited after being permitted) |
| Protection settings | Protection settings specify the lock bits for all blocks of code flash memory and the reset vector value for the function of variable reset vector. | Δ (Reset vector value for the function of variable reset vector is not supported.) | √ |
| Setting of option bytes | Option bytes are set to change them from the initial values for the RH850/P1M-E. | √ | √ |
| Clearing the configuration | ID setting, security settings, protection settings, and option byte settings are initialized. | √ | — |

For details on serial programming, see the user's manual of the flash programmer.

For details on self-programming, see the user's manuals for the flash memory hardware interface.

The flash memory supports various security functions.

The OTP setting and authentication of the ID code are security functions for use with serial programming and self-programming.

In serial programming, authentication of the ID code, prohibiting connection of a dedicated flash memory programmer, and prohibition of commands (for block erasure, programming, and reading) are available for use as security functions.

The security functions supported by the flash memory are listed in **Table 35.4**.

For details on security function, see the *RH850/P1M-E ICUSE User's Manual*.

Table 35.4 Summary of Security Functions

| Function | Description |
|--|---|
| OTP | OTP can be individually set for each block of the user area and the extended user area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self-programming is prohibited. Once set, the OTP setting cannot be released. Furthermore, since execution of the configuration clearing command is prohibited for any area for which OTP has been set, changing a security setting from "prohibited" to "permitted" is not possible. |
| ID authentication | The result of ID authentication can be used to control the connection of a dedicated flash memory programmer for serial programming. The result of ID authentication can also be used to control enabling of self-programming. |
| Prohibition of connection of a dedicated flash memory programmer | The connection of a dedicated flash memory programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a dedicated flash memory programmer is prohibited, changing a security setting from "prohibited" to "permitted" is not possible. |
| Prohibition of block erasure commands | Block erasure commands at the time of serial programming are prohibited. Since execution of the configuration clearing command is also prohibited when block erasure commands are prohibited, changing a security setting from "prohibited" to "permitted" is not possible. |
| Prohibition of programming commands | Block erasure commands and programming commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted. |
| Prohibition of read commands | Read commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted. |

Table 35.5 Available Operations and Security Settings

| Function | All Security Settings and Erasure, Programming, and Read Operations (√: Executable, x: Not Executable, —: Not Supported) | | Point for Caution Regarding the Security Setting | |
|--|--|---|---|--|
| | Serial programming | Self-programming | Serial programming | Self-programming |
| OTP | <ul style="list-style-type: none"> Areas for which OTP is set <ul style="list-style-type: none"> - Block erasure commands: x - Programming commands: x - Read commands: √ Areas for which OTP is not set <ul style="list-style-type: none"> - Block erasure commands: √ - Programming commands: √ - Read commands: √ | <ul style="list-style-type: none"> Areas for which OTP is set <ul style="list-style-type: none"> - Block erasure: x - Programming: x - Reading: √ Areas for which OTP is not set <ul style="list-style-type: none"> - Block erasure: √ - Programming: √ - Reading: √ | <ul style="list-style-type: none"> The OTP setting cannot be released. Execution of the configuration clearing command is not possible. | The OTP setting cannot be released. |
| ID authentication | <ul style="list-style-type: none"> When the ID codes do not match <ul style="list-style-type: none"> - Block erasure commands: x - Programming commands: x - Read commands: x When the ID codes match <ul style="list-style-type: none"> - Block erasure commands: √ - Programming commands: √ - Read commands: √ | <ul style="list-style-type: none"> When the ID codes do not match <ul style="list-style-type: none"> - Code flash memory <ul style="list-style-type: none"> - Block erasure: x - Programming: x - Reading: √ - Data flash memory <ul style="list-style-type: none"> - Block erasure: √ - Programming: √ - Reading: √ When the ID codes match <ul style="list-style-type: none"> - Block erasure: √ - Programming: √ - Reading: √ | <ul style="list-style-type: none"> The configuration clearing command can initialize the setting for prohibition. The setting for prohibition of block erasure commands is not available. The setting for prohibition of programming commands is not available. The setting for prohibition of read commands is not available. | ID authentication is always in effect. |
| Prohibition of the connection of a serial programmer | <ul style="list-style-type: none"> Block erasure commands: x Programming commands: x Read commands: x | <ul style="list-style-type: none"> Block erasure: √ Programming: √ Reading: √ | Since execution of the configuration clearing command is not supported, initialization of the setting for prohibition is not possible. | Initialization of the settings prohibited is impossible because the configuration clearing command is not supported. |
| Prohibition of block erasure commands | <ul style="list-style-type: none"> Block erasure commands: x Programming commands: √ Read commands: √ | <ul style="list-style-type: none"> Block erasure: √ Programming: √ Reading: √ | <ul style="list-style-type: none"> Since execution of the configuration clearing command is not supported, initialization of the setting for prohibition is not possible. The setting for prohibition of serial programmer connection is not available. The setting for ID authentication to be effective for serial programming is not available. | Initialization of the settings prohibited is impossible because the configuration clearing command is not supported. |
| Prohibition of programming commands | <ul style="list-style-type: none"> Block erasure commands: x Programming commands: x Read commands: √ | <ul style="list-style-type: none"> Block erasure: √ Programming: √ Reading: √ | <ul style="list-style-type: none"> Executing the configuration clearing command only can initialize the settings prohibited. | Initialization of the settings prohibited is impossible because the configuration clearing command is not supported. |
| Prohibition of read commands | <ul style="list-style-type: none"> Block erasure commands: √ Programming commands: √ Read commands: x | <ul style="list-style-type: none"> Block erasure: √ Programming: √ Reading: √ | <ul style="list-style-type: none"> The setting for ID authentication to be effective for serial programming is not available. | |

The flash memory supports various protection functions. The protection functions supported by the flash memory are listed in **Table 35.6**.

Table 35.6 Summary of Protection Functions

| Function | Description |
|-----------------------|---|
| Block protection | Lock bit settings can be individually made to enable or disable programming and erasure of each block of the user area and the extended user area of code flash memory. Programming and erasure by self-programming of an area for which the lock bit is set and the lock bit function is enabled are prohibited. Programming or erasure can proceed again when the lock bit function is disabled after having been enabled. When a block of code flash memory is erased, the lock bit for that block is also erased. |
| Hardware protection | The level on the FLMD pin can be set to prohibit programming and erasure of the code flash memory. - FLMD0 = 0: Programming prohibited - FLMD0 = 1: Programming permitted |
| Variable reset vector | The protection settings include control of the reset vector. As shown in Figure 35.4 , after programming of a new boot program while leaving the existing boot program in place, changing the reset vector is a safe way to change to the area holding the new boot program. The areas that can be specified by using the reset vector are the user area and the extended user area. |

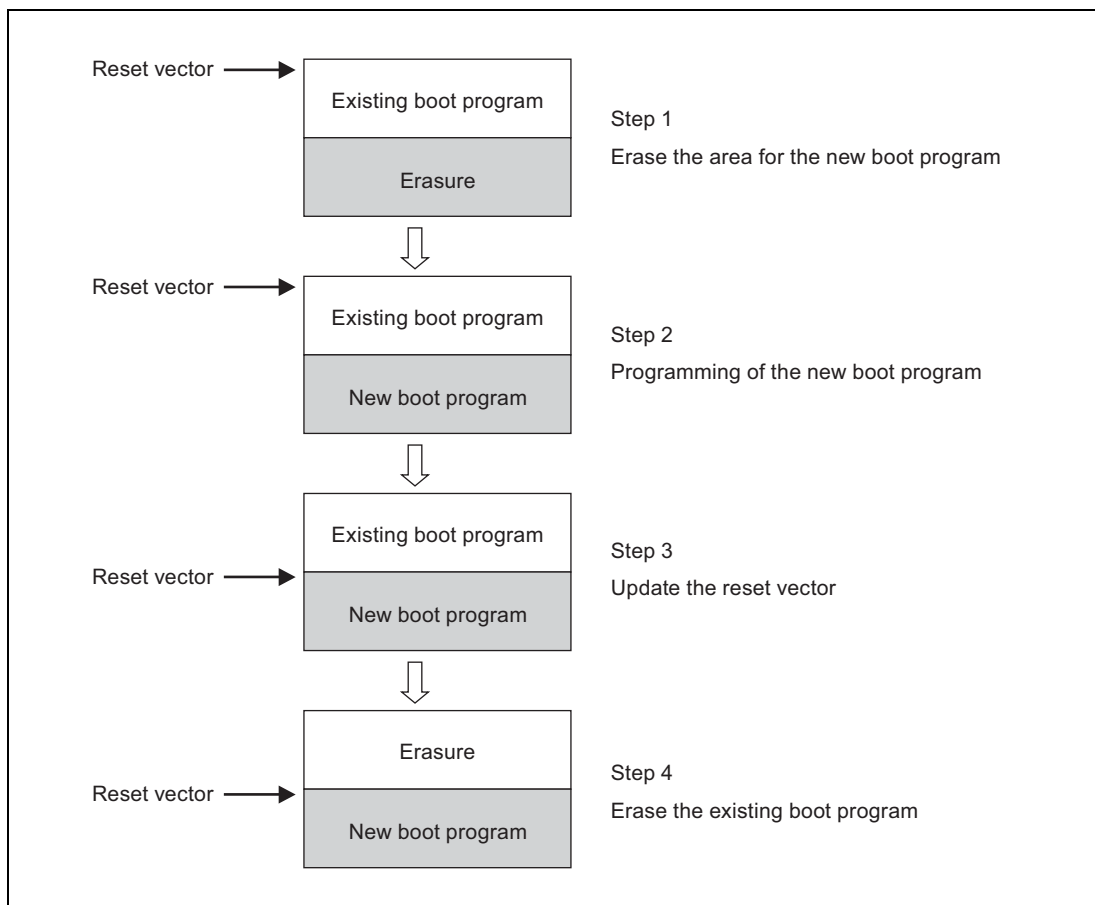


Figure 35.4 Utilizing the Variable Reset Vector Function to Update the Boot Program

NOTE

After step 4, a reset leads to updating of the reset vectors for which RBASE is changed.

35.5 Serial Programming

A dedicated flash memory programmer can be used to handle flash memory in serial programming mode.

35.5.1 Environments for Programming

The recommended environments for handling the flash memory of the microcontroller with data are described below.

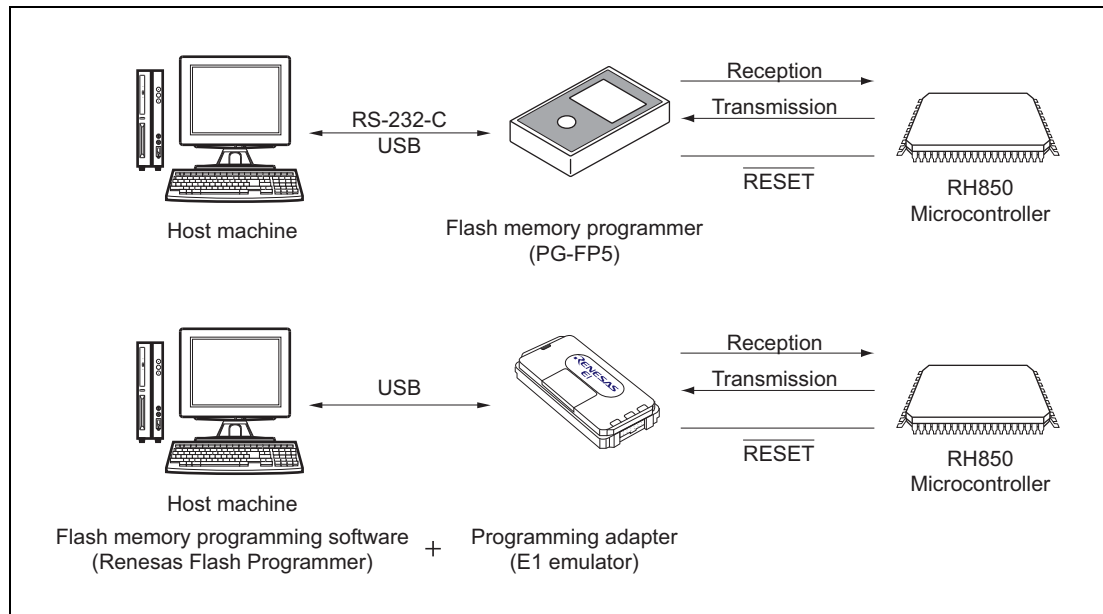


Figure 35.5 Flash Memory Programming Environment

By using the PG-FP5 flash memory programmer or the combination of the Renesas Flash Programmer (software for writing to flash memory) running on the host machine and the E1 emulator as an programming adaptor, the user is easily able to erase, program, and verify the contents of the on-chip memory of flash-memory-equipped microcontrollers from Renesas Electronics.

The PG-FP5 flash memory programmer handles programming from a host machine or programming in stand-alone mode while the Renesas Flash Programmer only handles programming from a host machine.

NOTE

For details on the PG-FP5, see the *PG-FP5 Flash Memory Programmer User's Manual*. For details on the Renesas Flash Programmer of flash programming software, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

35.6 Communication Modes

35.6.1 Asynchronous Flash Programming Interface — 1-Wire UART

The single-wire asynchronous serial programming interface, 1-wire UART is connected to the flash memory programmer with the following ports.

- FPDR(JP0_0): Receive data input/transmit data output

35.6.2 Asynchronous Flash Programming Interface — 2-Wire UART

The double-wire asynchronous serial programming interface, 2-wire UART is connected to the flash memory programmer with the following ports.

- FPDR(JP0_0): Receive data input
- FPDT(JP0_1): Transmit data output

35.6.3 Synchronous Flash Programming Interface CSI

The synchronous serial programming interface CSI is connected to the flash memory programmer with the following ports.

- FPDR(JP0_0): Receive data input
- FPDT(JP0_1): Transmit data output
- FPCK(JP0_2): Serial clock input

The flash memory programmer outputs the serial data clock SCK and the microcontroller operates as a slave.

NOTE

For details on Renesas Flash Programmer, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

35.6.4 Selection of Communication Method

In RH850/P1M-E, communication method can be selected by pulse input to the FLMD0 pin (up to 7 pulses) after transition to the flash memory programming mode. The FLMD0 pulse is generated by a dedicated flash memory programmer.

Figure 35.6 shows the relation between the number of pulses and communication method.

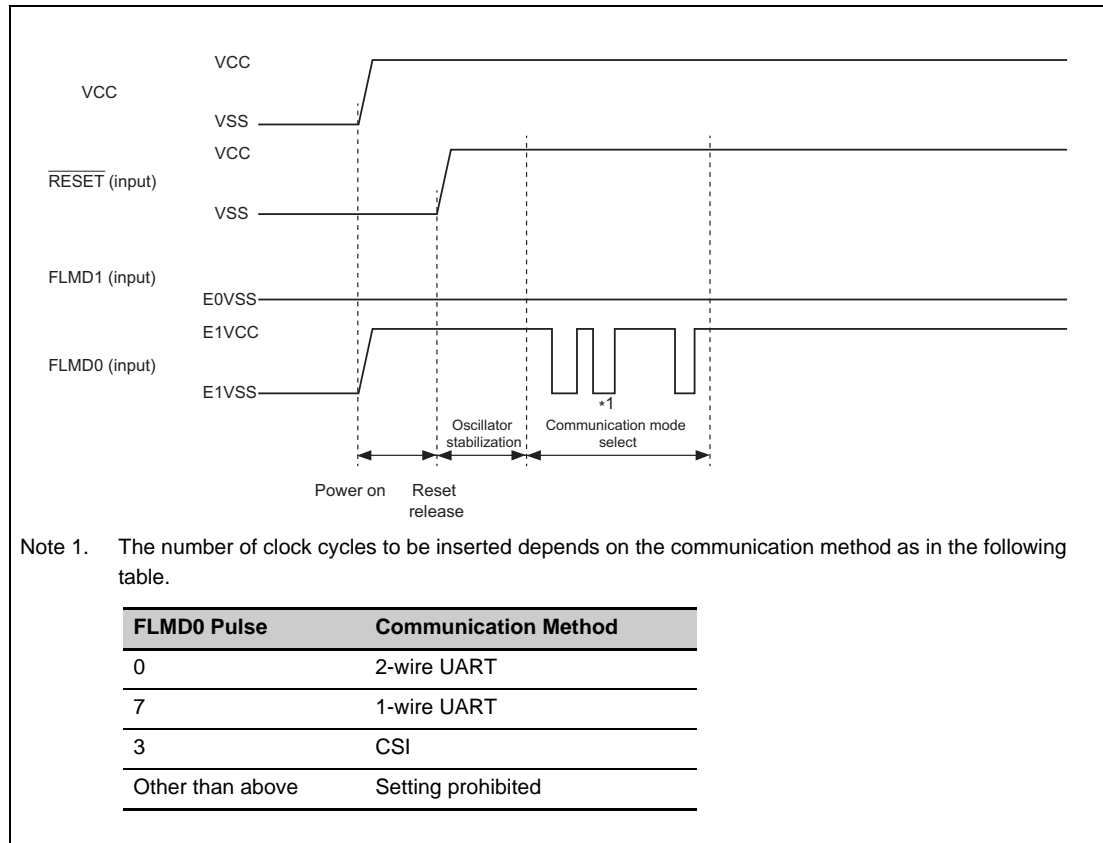


Figure 35.6 Selection of Communication Method

35.7 Self-Programming

35.7.1 Outline

The RH850/P1M-E supports programming of the flash memory by the user program itself.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to local RAM in advance of the programming operation, and executed from the given destination to perform the programming.

The programming program can be copied to the local RAM in advance and executed to program the code flash memory.

35.7.2 Background Operation

Background operations can be used when the combination of the flash memory for writing and the flash memory for reading is any of those listed below.

Table 35.7 Conditions under which Background Operation is Usable

| Range for Writing | Range for Reading |
|-------------------|-------------------|
| Data flash memory | Code flash memory |

35.7.3 Enabling Self-Programming

The self-programming function can be activated in normal operating mode.

Erase and programming of the code flash memory by the self-programming function is enabled by making the FLMD0 pin high level.

This prevents unnecessary overwriting of the program if the device operates incorrectly.

The FLMD0 pin is made high level by using one of the following methods.

- The FLMD0 pin is externally pulled up.
- The FLMD0 pin is pulled up by the FLMDCNT register.

The outline of the FLMDCNT register is described in **Section 35.7.3.1, FLMDCNT Register**.

35.7.3.1 FLMDCNT Register

This register specifies the internal pull-up or pull-down of the FLMD0 pin.

The correct write sequence using the FLMDPCMD register is required in order to update this register. For details, see **35.7.4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFA0 0000_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FLMDP UP |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 35.8 FLMDCNT Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 31 to 1 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 0 | FLMDPUP | FLMD0 Pin Software Control 0: Pull-down selected 1: Pull-up selected |

35.7.4 Write-Protected Registers

FLMDCNT register is a write-protected register.

This section describes how to write to write-protected registers.

Table 35.9 shows the list of write-protected registers.

Table 35.9 List of Write-protected Registers

| Module Name | Address | Register Name | Description | Access Size | Value after Reset | Protection Register | |
|-------------|-----------------------|------------------|-----------------------|-------------|------------------------|---------------------|-----------------|
| | | | | | | Command Register | Status Register |
| FLMD | FFA00000 _H | FLMDCNT Register | FLMD Control Register | 32 | 0000 0000 _H | FLMDPCMD | FLMDPS |

35.7.4.1 Write Procedure to Write-Protected Registers

Write access to a write-protected register is enabled by using the following write sequence:

1. Write the fixed value 0000 00A5_H to the protection command register.
2. Write the desired value to the protected register.
3. Write the inverse of the desired bit value to the protected register.
Write the inverse of the desired bit value to the reserved bits of the protected register.
4. Write the value written in step 2 to the protected register.
5. Verify that the desired value has been written to the protected register.
Verify successful write of the desired value to the protected register by verifying that the error monitor bit in the write sequence status register is “0”.
In case the write was not successful, indicated by the error monitor bit set to “1”, the entire sequence has to be restarted from step 1.

In case of any write access to a different register during step 1 to step 4 of the above sequence, the protection mechanism behaves as follows. (The same mechanism takes place when an interrupt is accepted during write sequence and write access to a different register is made during interrupt processing.)

- When write access to a different register is made in the same module, the write to the write-protection target register fails and the error monitor bit is set to 1.
- When write access to a register in another module is made, the write to the write protection register can be completed successfully.

For read access to a different register in the same module or a register in a different module is made, the write sequence does not fail.

35.7.4.2 Interrupt during Write Sequence

If an interrupt is generated during the write sequence, the protection mechanism operates as follows:

- (1) If an interrupt request is accepted during the write sequence and write access to a register of the same module is made:

The write operation to the write-protected register fails and the error monitor bit is set to 1. **Figure 35.7** shows an execution example.

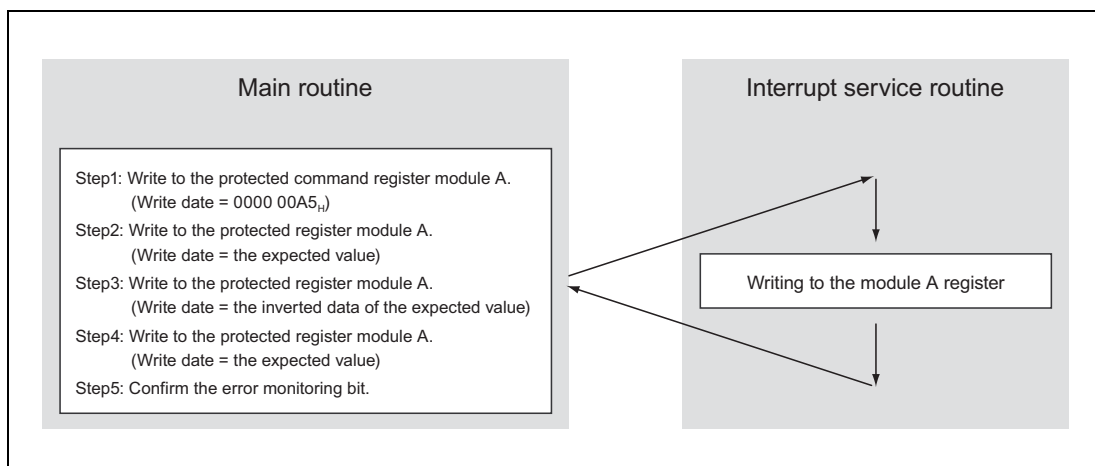


Figure 35.7 Example of Suspended Write Sequence

- (2) If an interrupt request is accepted during the write sequence and write access to a register of a different module is made.

The write operation to the write-protected register can be completed. **Figure 35.8** shows an execution example.

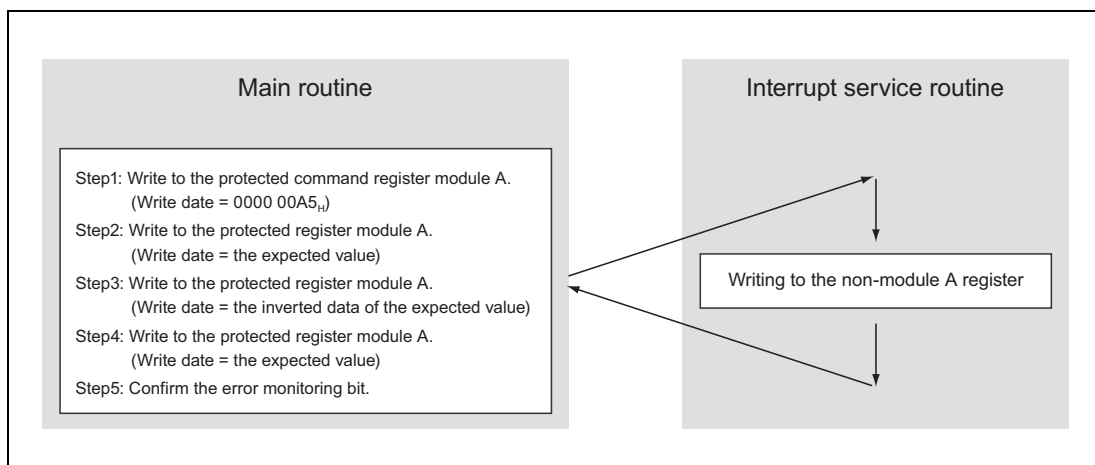


Figure 35.8 Example of Successful Write Sequence

For the write protection target registers of the RH850/P1M-E, see **Section Table 35.9, List of Write-protected Registers.**

35.7.4.3 Emulation Break during Write Sequence

If an emulation break occurs during the above write sequence, e.g. because of a breakpoint hit, the register protection is suspended until normal operation is resumed. Even if any register of the same module is accessed during the break, the write sequence is not suspended and the error monitor bit is not set to 1.

35.7.4.4 FLMDCNT Register Protection

(1) FLMDPCMD — FLMD Write Protection Command Register

This register is used as a command register to write to write-protection target registers.

Access: This register can only be written in 32-bit units.

Address: FFA0 0004_H

Value after reset: 0000 0000_H
This register is initialized by any reset source.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|-------------|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | FLMDPC[7:0] | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | W | W | W | W | W | W | W | W |

Table 35.10 FLMDPCMD Register Contents

| Bit Position | Bit Name | Function |
|--------------|-------------|---|
| 31 to 8 | Reserved | When writing, write 0. |
| 7 to 0 | FLMDPC[7:0] | Protection command that enables writing to write-protection target registers. |

(2) FLMDPS — FLMD Write Sequence Status Register

This register is used to indicate the status of write sequence of the write-protection target registers.

Access: This register can only be read in 32-bit units.

Address: FFA0 0008_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FLMDP RERR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 35.11 FLMDPS Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|--|
| 31 to 1 | Reserved | When read, the value after reset is read. |
| 0 | FLMDPRERR | Write Sequence Error Monitor 0: A protection error does not occur. 1: A protection error does occur. |

35.8 Reading Flash Memory

35.8.1 Reading Code Flash Memory

Special settings are not required to read code flash memory in normal mode. Data can simply be read out through access to addresses in the code flash memory.

Reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Furthermore, since the values of data cannot be guaranteed when an ECC error has been generated, use blank checking when you need to confirm that an area is in the non-programmed state.

NOTE

For details on blank checking, see the "8.5 Blank Checking of Code Flash Memory" of the *RH850/P1M-E Flash Memory User's Manual: Hardware Interface*.

35.8.2 Reading Data Flash Memory

Configure the number of read cycles in the EEPRDCYCL register prior to reading data from data flash memory in normal mode. Once this register is properly configured, data can be read by simply accessing addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programmed again are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

35.9 Description of Registers

35.9.1 Registers Related to Data Flash Memory

Table 35.12 shows the list of registers related to data flash memory.

Table 35.12 List of Registers Related to Data Flash Memory

| Register Name | Abbreviation | R/W | Value after Reset | Address | Access Size |
|---|--------------|-----|-------------------|------------------------|-------------|
| Data flash memory read cycle setting register | EEPRDCYCL | R/W | 0F _H | FFC5 9810 _H | 8 |

35.9.1.1 EEPRDCYCL — Data Flash Memory Read Cycle Setting Register

This register sets the read cycle of data flash memory.

This register is initialized by Power On Reset, System Reset 1, System Reset 2, or Application Reset 1.

Access: This register can be read/written in 8-bit units.

Address: FFC5 9810_H

Value after reset: 0F_H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---------------|-----|-----|-----|
| | — | — | — | — | FRDCYCLD[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W |

Table 35.13 EEPRDCYCL Register Contents

| Bit Position | Bit Name | Function |
|----------------------|---------------|---|
| 7 to 4 | Reserved | When read, the value after reset is returned. When writing, write the value after reset. |
| 3 to 0 | FRDCYCLD[3:0] | Number of data flash memory read cycles. Data flash memory is read in setting value + 1 cycle. |
| FRDCYCLD[3:0] | | Number of data flash memory read cycles |
| 0000 to 0010 | | Setting prohibited |
| 0011 | | 4 |
| 0100 | | 5 |
| 0101 | | 6 |
| 0110 | | 7 |
| 0111 | | 8 |
| 1000 | | 9 |
| Other than above | | 10 |

35.9.2 Registers Related to Write and Erase Protect of Flash Memory

Table 35.14 shows the list of registers related to write and erase protect of flash memory.

Table 35.14 Registers Related to Write and Erase Protect of Flash Memory

| Register Name | Abbreviation | R/W | Value after Reset | Address | Access Size |
|-------------------------|--------------|-----|------------------------|------------------------|-------------|
| FHVE15 control register | FHVE15 | R/W | 0000 0000 _H | FFF8 A430 _H | 32 |
| FHVE3 control register | FHVE3 | R/W | 0000 0000 _H | FFF8 2410 _H | 32 |

35.9.2.1 FHVE15 — FHVE15 Control Register

FHVE15 is a readable/writable register for software protection of flash memory against programming, erasure, and blank checking. Set both the FHVE15 and FHVE3 registers in the programmable, erasable, and blank-checkable state (0000 0001_H) to program, erase, or blank-check flash memory.

This register is initialized by Power On Reset, System Reset 1, System Reset 2, or Application Reset 1.

Access: This register can be read/written in 32-bit units.

Address: FFF8 A430_H

Value after reset: 0000 0000_H

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FHVE15 CNT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 35.15 FHVE15 Register Contents

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | FHVE15CNT | 0: Programming, erasure, and blank checking are disabled. 1: Programming, erasure, and blank checking are enabled. |

35.9.2.2 FHVE3 — FHVE3 Control Register

FHVE3 is a readable/writable register for software protection of flash memory against programming, erasure, and blank checking. Set both the FHVE15 and FHVE3 registers in the programmable, erasable, and blank-checkable state (0000 0001_H) to program, erase, or blank-check flash memory.

This register is initialized by Power On Reset, System Reset 1, or System Reset 2.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2410_H

Value after reset: 0000 0000_H

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FHVE3 CNT |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 35.16 FHVE3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 1 | Reserved | When read, the value after reset is read. When writing, write the value after reset. |
| 0 | FHVE3CNT | 0: Programming, erasure, and blank checking are disabled. 1: Programming, erasure, and blank checking are enabled. |

35.9.3 Registers Related to Product Information

Table 35.17 shows the list of registers related to product information.

Table 35.17 List of Registers Related to Product Information

| Register Name | Abbreviation | R/W | Value after Reset | Address | Access Size |
|-----------------------------------|--------------|-----|-------------------|------------------------|-------------|
| Product name storage register (1) | PRDNAME1 | R | See Table 35.19 | FFCD 00D0 _H | 32 |
| Product name storage register (2) | PRDNAME2 | R | See Table 35.19 | FFCD 00D4 _H | 32 |
| Product name storage register (3) | PRDNAME3 | R | See Table 35.19 | FFCD 00D8 _H | 32 |
| Product name storage register (4) | PRDNAME4 | R | See Table 35.19 | FFCD 00DC _H | 32 |

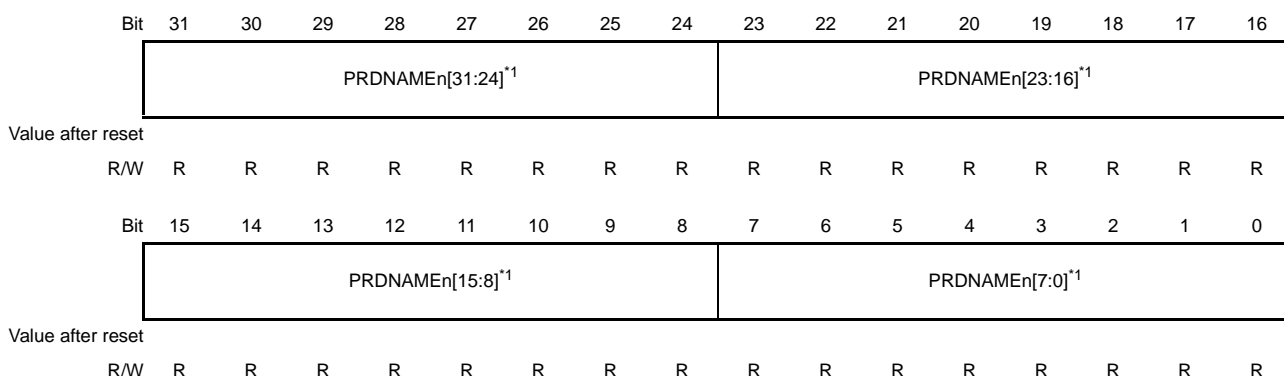
35.9.3.1 PRDNAME_n — Product Name Storage Register (n = 1 to 4)

This register stores the product name. The product model name is stored in 16-byte ASCII code, and PRDNAME1, PRDNAME2, PRDNAME3 and PRDNAME4 correspond to the fourth to first bytes, eighth to fifth bytes, twelfth to ninth bytes, and 16th to 13th bytes of the product model name respectively.

Access: This register can be read in 32-bit units.

Address: PRDNAME1: FFCD 00D0_H
 PRDNAME2: FFCD 00D4_H
 PRDNAME3: FFCD 00D8_H
 PRDNAME4: FFCD 00DC_H

Value after reset: See Table 35.19.



Note 1. n = 1 to 4

Table 35.18 PRDNAME_n Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|---|
| 31 to 24 | — | Product name fourth byte (PRDNAME1), eighth byte (PRDNAME2), twelfth byte (PRDNAME3), 16th byte (PRDNAME4) |
| 23 to 16 | — | Product name third byte (PRDNAME1), seventh byte (PRDNAME2), eleventh byte (PRDNAME3), 16th byte (PRDNAME4) |
| 15 to 8 | — | Product name second byte (PRDNAME1), sixth byte (PRDNAME2), tenth byte (PRDNAME3), 16th byte (PRDNAME4) |
| 7 to 0 | — | Product name first byte (PRDNAME1), fifth byte (PRDNAME2), ninth byte (PRDNAME3), 16th byte (PRDNAME4) |

Table 35.19 to list registers related to product information.

Table 35.19 List of Registers Related to Product Information

| Product Model Name | PRDNAME1 | PRDNAME2 | PRDNAME3 | PRDNAME4 |
|--------------------|-----------|-----------|-----------|-----------|
| R7F701375 | 3746 3752 | 3733 3130 | 2020 2035 | 2020 2020 |
| R7F701376 | 3746 3752 | 3733 3130 | 2020 2036 | 2020 2020 |
| R7F701377 | 3746 3752 | 3733 3130 | 2020 2037 | 2020 2020 |
| R7F701378 | 3746 3752 | 3733 3130 | 2020 2038 | 2020 2020 |
| R7F701379 | 3746 3752 | 3733 3130 | 2020 2039 | 2020 2020 |
| R7F701380 | 3746 3752 | 3833 3130 | 2020 2030 | 2020 2020 |
| R7F701381 | 3746 3752 | 3833 3130 | 2020 2031 | 2020 2020 |
| R7F701382 | 3746 3752 | 3833 3130 | 2020 2032 | 2020 2020 |
| R7F701383 | 3746 3752 | 3833 3130 | 2020 2033 | 2020 2020 |
| R7F701384 | 3746 3752 | 3833 3130 | 2020 2034 | 2020 2020 |
| R7F701385 | 3746 3752 | 3833 3130 | 2020 2035 | 2020 2020 |
| R7F701386 | 3746 3752 | 3833 3130 | 2020 2036 | 2020 2020 |

35.9.4 Registers Related to Reset Vector

Table 35.20 shows the list of registers related to reset vector.

Table 35.20 List of Registers Related to Reset Vector

| Register Name | Abbreviation | R/W | Value after Reset | Address | Access Size |
|-------------------------|--------------|-----|-----------------------|------------------------|-------------|
| Reset vector 0 register | GREG8 | R | Specified by the user | FFCD 0020 _H | 32 |

35.9.4.1 GREG8 — Reset Vector 0 Register

Access: This register can be read in 32-bit units.

Address: FFCD 0020_H

Value after reset: Specified by the user

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | Reset Vector 0 | | | | | | | | | | | | | | | |
| Value after reset | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | Reset Vector 0 | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 | 1/0 |

Table 35.21 GREG8 Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------------|--------------|
| 31 to 0 | Reset Vector 0 | Reset Vector |

35.10 Option Bytes

The option bytes of the flash memory are an expansion area and hold data specified by the user for a variety of purposes. Initial settings for peripheral modules and so on as specified by the option bytes become effective on release from the reset state.

35.10.1 Option Byte Setting

Be sure to set the option byte area that corresponds to the optional functions listed below, before writing a program to the flash memory.

The optional functions specified by the option bytes are as follows.

- Function of port group JP0
- Activation code method of WDTA0
- Start mode of WDTA0
- Enabling or disabling WDTA0
- Initial value of the overflow interval time for WDTA0
- Setting of $\overline{\text{EVTO}}/\overline{\text{EVTI}}$
- $\overline{\text{ERROROUT}}$ output mode setting

35.10.1.1 Setting of Option Byte 0

| | | | | | | | | | | | | | | | | |
|-----|---------|--------|----|----|--------------|----|----|----|----|---------|---------|----|----|----|-----------|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | OPWDRUN | — | — | — | OPWDOVF[2:0] | | | — | — | OPWDVAC | OPWDMDS | — | — | — | — | — |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OPEVTO | OPEVTI | — | — | — | — | — | — | — | — | — | — | — | — | ERROUTSEL | — |

Table 35.22 Option Byte 0 Setting

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|--------------|--|----------------------------|----------|----------|---------------|---|---|---|-------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|
| 31 | OPWDRUN | Specifies the start mode for WDTA0. 0: Software trigger start mode 1: Default start mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 30 to 28 | Reserved | When writing, write 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 27 to 25 | OPWDOVF[2:0] | Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0]. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>OPWDOVF2</th> <th>OPWDOVF1</th> <th>OPWDOVF0</th> <th>Overflow Time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>$2^9 / \text{WDTATCKI}$</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>$2^{10} / \text{WDTATCKI}$</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>$2^{11} / \text{WDTATCKI}$</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>$2^{12} / \text{WDTATCKI}$</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>$2^{13} / \text{WDTATCKI}$</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>$2^{14} / \text{WDTATCKI}$</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>$2^{15} / \text{WDTATCKI}$</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>$2^{16} / \text{WDTATCKI}$</td> </tr> </tbody> </table> | OPWDOVF2 | OPWDOVF1 | OPWDOVF0 | Overflow Time | 0 | 0 | 0 | $2^9 / \text{WDTATCKI}$ | 0 | 0 | 1 | $2^{10} / \text{WDTATCKI}$ | 0 | 1 | 0 | $2^{11} / \text{WDTATCKI}$ | 0 | 1 | 1 | $2^{12} / \text{WDTATCKI}$ | 1 | 0 | 0 | $2^{13} / \text{WDTATCKI}$ | 1 | 0 | 1 | $2^{14} / \text{WDTATCKI}$ | 1 | 1 | 0 | $2^{15} / \text{WDTATCKI}$ | 1 | 1 | 1 | $2^{16} / \text{WDTATCKI}$ |
| OPWDOVF2 | OPWDOVF1 | OPWDOVF0 | Overflow Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | $2^9 / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | $2^{10} / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | $2^{11} / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | $2^{12} / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | $2^{13} / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | $2^{14} / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | $2^{15} / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | $2^{16} / \text{WDTATCKI}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24, 23 | Reserved | When writing, write 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | OPWDVAC | Specifies the trigger register for generating a counter restart trigger to avoid counter overflow. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) Note: For details, see Table 21.7, WDTA Start-up Options. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | OPWDMDS | Selects the counter clock source of WDTA0. 0: High-speed mode (CLK_IOSC): 8 MHz 1: Low-speed mode (CLK_IOSC/32): 250 kHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 to 16 | Reserved | When writing, write 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | OPEVTO | EVTO Setting 0: EVTO is used 1: EVTO is not used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | OPEVTI | EVTI Setting 0: EVTI is used 1: EVTI is not used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 to 2 | Reserved | When writing, write 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | ERROUTSEL | ERROUTSEL 0: ERROROUT signals from master and checker are output separately. 1: ERROROUT signals from master and checker are output after merged. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Reserved | When writing, write 1. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

35.10.1.2 Setting of Option Byte 2

| | | | | | | | | | | | | | | | | |
|-----|----|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | OPJTAG[1:0] | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

Table 35.23 Option Byte 2 Setting

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | |
|--------------|-------------|---|---------|---------|-----------------|---|---|------|---|---|--------------|---|---|--------------------|---|---|--------------|
| 31 | Reserved | When writing, write 1. | | | | | | | | | | | | | | | |
| 30, 29 | OPJTAG[1:0] | Switch of the Debug Interfaces The following debug interface is selected depending on the combination of the OPJTAG1 and OPJTAG0. <table border="1" data-bbox="678 792 1417 981"> <thead> <tr> <th>OPJTAG1</th> <th>OPJTAG0</th> <th>Debug Interface</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>GPIO</td> </tr> <tr> <td>0</td> <td>1</td> <td>LPD (4 pins)</td> </tr> <tr> <td>1</td> <td>0</td> <td>Setting prohibited</td> </tr> <tr> <td>1</td> <td>1</td> <td>Nexus (JTAG)</td> </tr> </tbody> </table> | OPJTAG1 | OPJTAG0 | Debug Interface | 0 | 0 | GPIO | 0 | 1 | LPD (4 pins) | 1 | 0 | Setting prohibited | 1 | 1 | Nexus (JTAG) |
| OPJTAG1 | OPJTAG0 | Debug Interface | | | | | | | | | | | | | | | |
| 0 | 0 | GPIO | | | | | | | | | | | | | | | |
| 0 | 1 | LPD (4 pins) | | | | | | | | | | | | | | | |
| 1 | 0 | Setting prohibited | | | | | | | | | | | | | | | |
| 1 | 1 | Nexus (JTAG) | | | | | | | | | | | | | | | |
| 28 to 0 | Reserved | When writing, write 1. | | | | | | | | | | | | | | | |

35.10.2 OPBT0 — Option Byte 0

Access: This register can only be read in 32-bit units.

Address: FFCD 0030_H

Value after reset: Specified by the user

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-------------------|---------|--------|----|----|--------------|-----|-----|----|----|---------|---------|----|----|----|-----------|----|
| | OPWDRUN | — | — | — | OPWDOVF[2:0] | | | — | — | OPWDVAC | OPWDMDS | — | — | — | — | — |
| Value after reset | 0/1 | 1 | 1 | 1 | 0/1 | 0/1 | 0/1 | 1 | 1 | 0/1 | 0/1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OPEVTO | OPEVTI | — | — | — | — | — | — | — | — | — | — | — | — | ERROUTSEL | — |
| Value after reset | 0/1 | 0/1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0/1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 35.24 OPBT 0 Setting (1/2)

| Bit Position | Bit Name | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|--------------|---|----------------------------|----------|----------|----------------|---|---|---|---------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|---|---|---|----------------------------|
| 31 | OPWDRUN | Specifies the start mode for WDTA0. 0: Software trigger start mode 1: Default start mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 30 to 28 | Reserved | When read, the value after reset is returned. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 27 to 25 | OPWDOVF[2:0] | Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0]. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th>OPWDOVF2</th> <th>OPWDOVF1</th> <th>OPWDOVF0</th> <th>Over flow Time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>2⁹ / WDTATCKI</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>2¹⁰ / WDTATCKI</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>2¹¹ / WDTATCKI</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2¹² / WDTATCKI</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>2¹³ / WDTATCKI</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>2¹⁴ / WDTATCKI</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>2¹⁵ / WDTATCKI</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2¹⁶ / WDTATCKI</td> </tr> </tbody> </table> | OPWDOVF2 | OPWDOVF1 | OPWDOVF0 | Over flow Time | 0 | 0 | 0 | 2 ⁹ / WDTATCKI | 0 | 0 | 1 | 2 ¹⁰ / WDTATCKI | 0 | 1 | 0 | 2 ¹¹ / WDTATCKI | 0 | 1 | 1 | 2 ¹² / WDTATCKI | 1 | 0 | 0 | 2 ¹³ / WDTATCKI | 1 | 0 | 1 | 2 ¹⁴ / WDTATCKI | 1 | 1 | 0 | 2 ¹⁵ / WDTATCKI | 1 | 1 | 1 | 2 ¹⁶ / WDTATCKI |
| OPWDOVF2 | OPWDOVF1 | OPWDOVF0 | Over flow Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 2 ⁹ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 2 ¹⁰ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 2 ¹¹ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 2 ¹² / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 2 ¹³ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 2 ¹⁴ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 2 ¹⁵ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 2 ¹⁶ / WDTATCKI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Note: WDTATCKI is frequency of watchdog timer operation clock. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24, 23 | Reserved | When read, the value after reset is read. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22 | OPWDVAC | Specifies the trigger register for generating a counter restart trigger to avoid counter overflow. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) Note: for details, see Table 21.7, WDTA Start-up Options. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 21 | OPWDMDS | Selects the counter clock source of WDTA0. 0: High-speed mode (CLK_IOSC: 8 MHz) 1: Low-speed mode (CLK_IOSC/32: 250 kHz) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 to 16 | Reserved | When read, the value after reset is returned. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | OPEVTO | EVTO Setting 0: EVTO is used 1: EVTO is not used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | OPEVTI | EVTI Setting 0: EVTI is used 1: EVTI is not used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 to 2 | Reserved | When read, the value after reset is returned. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 35.24 OPBT 0 Setting (2/2)

| Bit Position | Bit Name | Function |
|--------------|-----------|---|
| 1 | ERROUTSEL | ERROUTSEL 0: <u>ERROROUT</u> signals from master and checker are output separately. 1: <u>ERROROUT</u> signals from master and checker are output after merged. |
| 0 | Reserved | When read, the value after reset is returned. |

35.10.3 OPBT2 — Option Byte 2

Access: This register can only be read in 32-bit units.

Address: FFCD 0038_H

Value after reset: Specified by the user

| | | | | | | | | | | | | | | | | |
|-------------------|----|-------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | OPJTAG[1:0] | | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 0/1 | 0/1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 35.25 OPBT 2 Setting

| Bit Position | Bit Name | Function | |
|--------------|-------------|--|--------------------|
| 31 | Reserved | When read, the value after reset is read. | |
| 30, 29 | OPJTAG[1:0] | Switch of the Debug Interfaces The following debug interface is selected depending on the OPJTAG[1:0] | |
| | OPJTAG1 | OPJTAG0 | Debug Interface |
| | 0 | 0 | GPIO |
| | 0 | 1 | LPD (4 pin) |
| | 1 | 0 | Setting prohibited |
| | 1 | 1 | Nexus (JTAG) |
| 28 to 0 | Reserved | When read, the value after reset is read. | |

35.12 Usage Note

1. Reading areas where programming or erasure was interrupted
When programming or erasure of an area of flash memory is interrupted, the data stored in the area become undefined. To avoid the reading out of undefined data, which might cause a malfunction, take care not to fetch instructions or read data from areas where programming or erasure was interrupted.
2. Reading the code flash memory that has been erased but not yet been programmed again
Note that reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Use blank checking when you need to confirm that an area is in the non-programmed state.

NOTE

For details on blank checking, see the "8.5 Blank Checking of Code Flash Memory" of the *RH850/P1M-E Flash Memory User's Manual: Hardware Interface*.

3. Prohibition of additional writing
Writing to a given area two or more times is not possible. When overwriting data in an area of flash memory after writing to the area has been completed, erase the area first.
4. Reset during programming and erasure
In the case of a reset due to the signal on the RESET pin during programming and erasure, wait for at least the minimum width of reset pulse once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal before releasing the device from the reset state.
5. Allocation of vectors for interrupts and other exceptions during programming and erasure
Generation of an interrupt or other exception during programming or erasure may lead to fetching of the vector from the code flash memory. If this does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory.
6. Abnormal termination of programming and erasure
Even if programming/erasure ends abnormally due to reset input or power off, the programming/erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming/erasure ends abnormally, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again to ensure that the corresponding area is completely erased before using.
If programming and erasure of code flash memory are not completed normally, the lock bit for the target area may be enabled (locked). In such cases, erase the block to erase the lock bit while the lock bit is in the disabled state (the area is not locked).
7. Items prohibited during programming and erasure and blank checking
Do not perform the following operations during programming and erasure and blank checking.
 - Set the operating voltage from the power supply outside the allowed range.
 - Update the value of FHVE15 and FHVE3.

Section 36 RAM

36.1 Features

This chip has the following RAMs:

- Instruction cache RAM for PE
- Local RAM (LRAM) for PE
- Global RAM (GRAM)
- Emulation RAM
- DTS RAM
- CSIH RAM
- FlexRay RAM
- RS-CANFD RAM

From the viewpoint of functional safety, all RAMs, except for Instruction cache RAM and ERAM, are protected by Error-Correcting Code (ECC). Instruction cache RAM provides only Error-Detecting Code (EDC).

36.1.1 List of On-chip RAM

Table 36.1 shows the list of on-chip RAM.

Table 36.1 List of On-chip RAM

| RAM List | P1M-E |
|------------------------------|--|
| Instruction cache RAM (tag) | 256words × 4 |
| Instruction cache RAM (data) | 4KB × 4 |
| Local RAM (PE1) | 128KB |
| Global RAM (Bank A/B) | 64KB |
| Emulation RAM | 32KB ^{*1} / 8KB ^{*2} |
| DTS RAM | 4KB |
| CSIH RAM | 512B/channel |
| FlexRay RAM | 8KB/module 512Bx2/module |
| RS-CANFD RAM | 3KB/module 7.4KB/module |

Note 1. Only available in devices with 2-MB flash memory

Note 2. Only available in devices with 1-MB flash memory

36.2 Overview

36.2.1 Function overview

36.2.1.1 Access

The CPU, DMAC/DTS, and H-Bus masters can access Local RAM (PE1) and Global RAM (Bank A/B). For details, see **Section 3, CPU System**.

36.2.1.2 Emulation RAM

The specific area of the code flash can be replaced with this RAM on a page unit basis. The access latency from the CPU after replacement is the same as that of the Code Flash.

36.2.1.3 ECC

From the viewpoint of functional safety, all RAM, except for Instruction cache RAM and Emulation RAM, are protected by Error-Correcting Code (ECC). Instruction cache RAM provides only Error-Detecting Code (EDC). For details, see **Section 31, Functional Safety**.

36.2.1.4 RAM initialization

To avoid time-consuming initialization by software, a hardware mechanism is implemented to initialize the following RAMs. This initialization includes correct setting of the related ECC bits.

- LRAM for PE
- GRAM
- DTS RAM
- CSIH RAM

RAM initialization to 0 is executed by all reset. RAM initialization can be disabled according to the setting in RAM Initialization Mode Control Registers. For details, see **Section 8, Reset Controller**.

36.3 Backup Register

36.3.1 List of Registers

The backup registers are listed in the following table.

Table 36.2 List of Registers

| Register Name | Symbol | Address |
|-------------------|----------|------------------------|
| Backup register 0 | BRAMDAT0 | FFC0 A000 _H |
| Backup register 1 | BRAMDAT1 | FFC0 A004 _H |
| Backup register 2 | BRAMDAT2 | FFC0 A008 _H |
| Backup register 3 | BRAMDAT3 | FFC0 A00C _H |

36.3.2 BRAMDAT_n — Backup Register

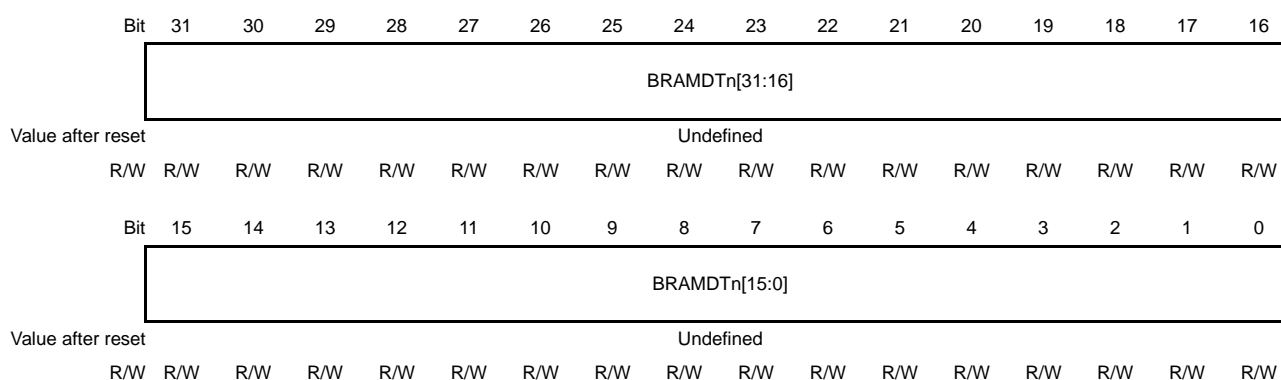
The BRAMDAT_n registers can be read/written in 32-bit units.

Storing data in backup registers before self-diagnosis (BIST) enables the retention of data during this test.

This register is not initialized by any reset.

Access: This register can be read/written in 32-bit units.

Address: FFC0 A000_H + n × 4_H (n = 0 to 3)



36.4 Emulation RAM

36.4.1 Emulation RAM

The RH850/P1M-E includes emulation RAM to emulate the code flash. The emulation RAM consists of 8 Kbytes (one 8-Kbyte bank) of flash memory for products with 1 Mbyte of flash memory, and 32 Kbytes (four 8-Kbyte banks) for products with 2 Mbytes of flash memory. The emulation RAM is available for the code flash emulation described below.

36.4.2 Code Flash Emulation Function Using the Emulation RAM

Mapping to the code flash area enables code flash to move to the emulation RAM and it enables the emulation RAM to emulate the code flash. The code flash data can be dynamically modified during execution of a user program via the emulation RAM which has been mapped to the code flash area.

Figure 36.1 shows the circuit configuration around the emulation RAM.

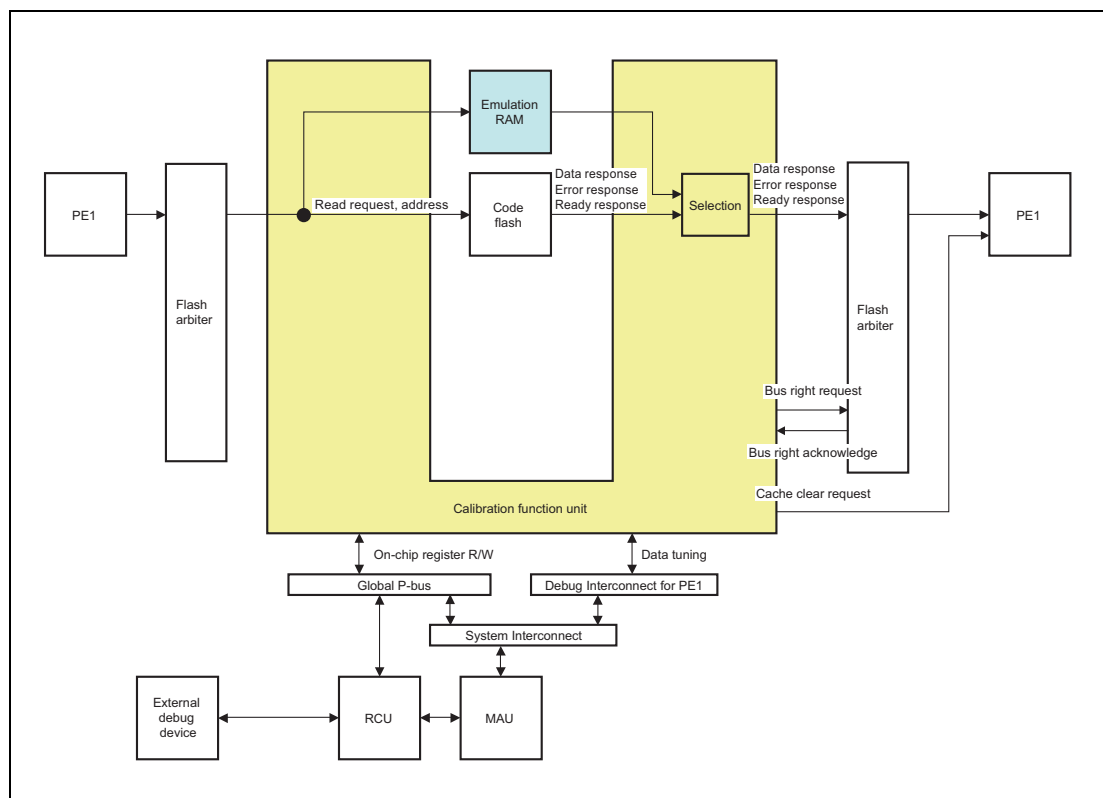


Figure 36.1 Circuit Around the Emulation RAM

36.4.3 EmulationRAM memory map

The ERAM memory map in RH850/P1M-E is shown below.

The ERAM area consists of four blocks of ERAM mapped at 128 KB intervals.

Each block contains 8 KB of ERAM, making a total of 32 KB.

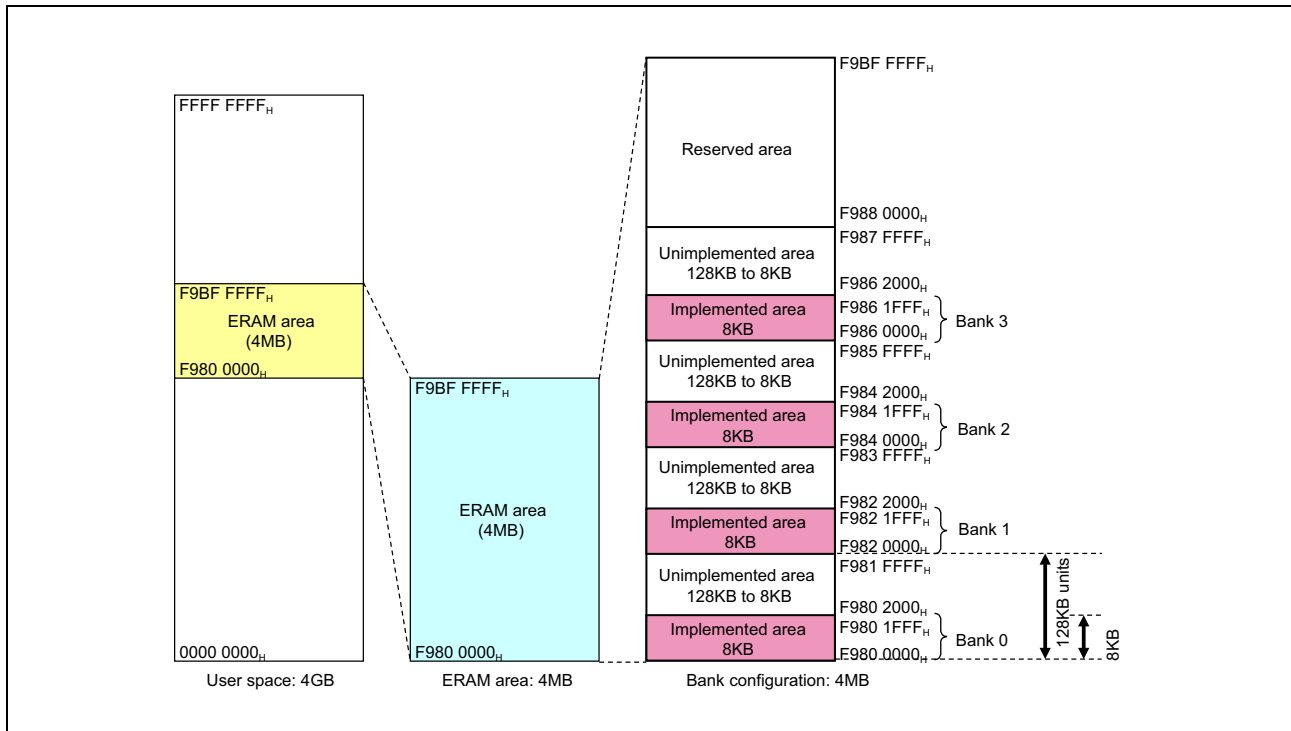


Figure 36.2 Emulation RAM memory map

36.5 Calibration Function

For the memory overlay and Emulation RAM functionality the Calibration Function Unit (CFU) is provided. This unit supports to configure overlay of Emulation RAM (ERAM) onto the Flash memory space accessed by PE1.

36.5.1 Calibration Function Unit (CFU)

The unit CFU is used to manage Flash overlay mechanism. Based on given configuration via debug control interface such as JTAG the unit manages to overlay RAM blocks onto Flash memory.

36.5.2 CFU Register List

Function registers of calibration function unit (CFU) are mapped to FFFF7800_H to FFFF79FF_H in the peripheral I/O area

Table 36.3 List of Registers

| Address | Register | Function |
|------------------------|----------|---------------------------------|
| FFFF 7808 _H | TM_CC | Cache clear operation |
| FFFF 7810 _H | TM_ME | ERAM bank 0 to 3 mapping enable |
| FFFF 7814 _H | TM_MS | ERAM bank 0 to 3 mapping status |
| FFFF 7840 _H | TM_MA0 | ERAM bank 0 mapping address |
| FFFF 7844 _H | TM_MA1 | ERAM bank 1 mapping address |
| FFFF 7848 _H | TM_MA2 | ERAM bank 2 mapping address |
| FFFF 784C _H | TM_MA3 | ERAM bank 3 mapping address |

36.5.3 TM_CC — Cache Clear Operation Register

This register is used to issue cache clear requests to the unit that has the flash memory data cache.

This register can be read or written in 32-bit units.

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CCLR |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W |

Table 36.4 TM_CC Register Contents

| Bit Position | Bit Name | Function |
|--|----------|---|
| 0 | CCLR | <p>A cache clearing request is issued by writing 1 to this bit. Writing 0 has no effect.</p> <p>The value read is 1 if this bit is read before the response to a cache clear request is completed and 0 after the response to a cache clear request is completed.</p> |
| <p>CAUTION</p> <p>After 1 has been written to this bit, writing 1 to the bit is ignored if this is done before the response to the earlier cache clear request is complete (the cache is not cleared in response to the latter request). After 1 has been written to this bit, if 0 is written before the response to the cache clear request is complete, this will not cancel the response to the earlier request. When executing cache clear requests in succession, it is necessary to check that the previous cache clear request is complete (by reading this bit and confirming that it is 0) before issuing the next cache clear request.</p> | | |
| <p>0: (When read) A cache clear request is not being processed or the response to a cache clear request is complete.</p> <p>1: (When written) Issue a cache clear request. (When read) A cache clear request is being processed.</p> | | |

36.5.4 TM_ME — Tuning Memory Mapping Enable Register

The tuning memory mapping enable register is used to control mapping of ERAM banks 0 to 3 to the flash area.

This register can be read or written in 32-bit units.

CAUTIONS

1. When the mapping enable bit TME_n is updated (including writing of its current value again (updating without changing the value)), a cache clear request is issued. (n = 0 to 3)
 2. Before ERAM bank n and tuning memory mapping address register n (the TM_MAn register) are updated, the corresponding mapping enable bit TME_n must be set to 0 (disabling mapping) (n = 0 to 3).
 3. After updating the TM_ME register, wait and verify that the value read from the TM_MS register is the same as the setting of the TM_ME register (that is, wait until the mapping setting specified in the TM_ME register is actually applied).
 4. After updating the mapping enable bit TME_n, the following processing is executed automatically and the mapping setting of each ERAM bank is switched.
 - (1) Updating mapping enable bits TME0 to TME3 causes the calibration function unit (CFU) to issue a flash memory bus right request to the flash arbiter.
 - (2) After acquiring the flash memory bus right, the calibration function unit (CFU) switches the mapping settings of all the ERAM banks.
 - (3) The calibration function unit (CFU) issues a cache clear request
(→to maintain coherency between the data in the flash memory or ERAM and the cache).
 - (4) The calibration function unit (CFU) releases the flash memory bus right. However, if the mapping enable bit TME_n is updated during a system reset while the debug circuit reset mask request signal is high (db_rmsk = “H”), the calibration function unit (CFU) immediately switches the mapping settings of the ERAM banks without acquiring the flash memory bus right.
 5. The mapping settings of the ERAM banks are switched independently of the CPU instruction execution sequence.
-

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----------|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | TME[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W |

Table 36.5 TM_ME Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 3 to 0 | TME[3:0] | These bits control mapping for the user space of ERAM bank n (flash area). (n = 0 to 3) 0: Mapping disabled. 1: Mapping enabled. |

CAUTION

In 1 MB flash memory products, only TME0 can be set to 1 (mapping enabled). Do not set TME1 to TME3 to 1 (mapping enabled).

36.5.5 TM_MS — Tuning Memory Mapping Status Register

The tuning memory mapping status register indicates the state of mapping for the flash area of ERAM banks 0 to 3.

This register is a read-only register that can be read in 32-bit units.

CAUTION

After updating the TM_ME register, wait and verify that the value read from the TM_MS register is the same as the setting of the TM_ME register (that is, wait until the mapping setting specified in the TM_ME register is actually applied).

| | | | | | | | | | | | | | | | | |
|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | MES[3:0] | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

Table 36.6 TM_MS Register Contents

| Bit Position | Bit Name | Function |
|--------------|----------|--|
| 3 to 0 | MES[3:0] | These bits indicate the state of mapping for the user space of ERAM bank n (flash area). (n = 0 to 3) 0: Mapping disabled. 1: Mapping enabled. |

36.5.6 TM_MA0 to 3 — Tuning Memory Mapping Address Registers 0 to 3

Tuning memory mapping address registers 0 to 3 are used to set the mapping addresses of ERAM banks 0 to 3.

ERAM banks 0 to 3 can be mapped to any address range (up to 4 areas) in the flash memory. The area to which ERAM bank n (n = 0 to 3) is mapped in the flash memory (ROM) can be replaced by ERAM bank n (n = 0 to 3) (this is known as the flash emulation function). This allows any area in the flash memory to be read or written.

Mapping addresses are set in units of 8 KB/bank.

Mapping of up to 4 banks is possible.

This register can be read or written in 32-bit units.

CAUTIONS

1. Before tuning memory mapping address register n (the TM_MAn register) is updated, the corresponding mapping enable bit TMEn must be set to 0 (disabling mapping). (n = 0 to 3)
2. Be sure to configure the external debugger so that the mapping settings (mapping address of the bank for which mapping is enabled) of ERAM banks 0 to 3 do not overlap. The CFU cannot access the ERAM correctly if the mapping settings are such that multiple banks are hit at the same time.

| | | | | | | | | | | | | | | | | |
|-------------------|-------------|-----|-----|-----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | TMAn[31:16] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TMAn[15:0] | | | | | | | | | | | | | | | |
| Value after reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R | R | R | R | R | R | R |

Table 36.7 TM_MA0 to 3 Register Contents

| Bit Position | Bit Name | Function |
|--------------|--------------|---|
| 31 to 0 | TMAn[31:0] | These bits specify the mapping address of ERAM bank n. CAUTION <ul style="list-style-type: none"> • Specify an address within the flash area (flash cache) in the user space. If an address outside the flash area (flash cache) is specified, the operation is not guaranteed. • The TMAn31 to 25 and 12 to 0 bits are fixed to “0”. |
| | Mapping unit | Bits that determine mapping |
| | 8 KB/block | TMAn24 to 13 |
| | | Ignored bits |
| | | TMAn31 to 25, 12 to 0 |

36.6 Usage Notes

On path between Local RAM and CPU, buffers are implemented to realize fast Local RAM access.

Therefore when a load instruction is executed for the same address after a store instruction to Local RAM, the load instruction may read out data from buffers instead of data on Local RAM. Either of following procedure is to surely read data on Local RAM, Either of following procedures can be used to surely read data on Local RAM.

1. Read out the first written data, only after more than 32 bytes of data are written into Local RAM.
2. Execute a SYNCM instruction before a load instruction is executed on the same address after a store instruction to the Local RAM.

Section 37 Electrical Specifications

37.1 Overview

The specifications in this section are for devices operating under the following conditions. Where a special condition is required for a given specification, the condition will be indicated. Furthermore, the specifications in this section are not guaranteed unless the conditions listed below are met.

37.1.1 General Measurement Conditions

37.1.1.1 Common Conditions

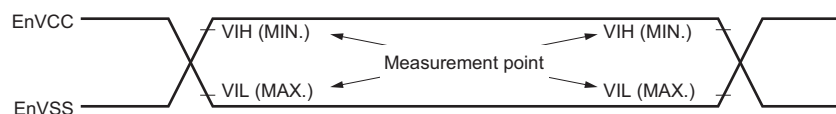
- Power supply
 - VCC = 3.0 V to 5.5 V
 - EnVCC = 3.0 V to 5.5 V
 - VDD = 1.20 V to 1.35 V^{*1}
 - A0VCC, A1VCC = 3.0 V to 5.5 V
 - A0VREFH = 3.0 V to 5.5 V
 - A1VREFH = 3.0 V to 5.5 V
 - EnVSS = VSS = A0VSS = A1VSS = 0 V
- Capacitance of the internal regulator
 - CVCL: 0.1 μ F \pm 30%^{*2}
- Operating temperature
 - Tj = -40 to +150°C
- Load conditions
 - CL = 30 pF

Note 1. Only applicable for DPS products

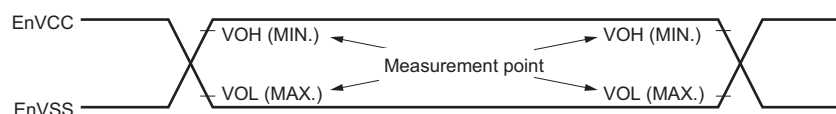
Note 2. See **Section 37.7, Regulator Characteristics.**

37.1.1.2 AC Characteristic Measurement Condition

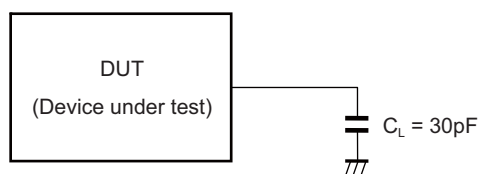
(1) AC test input measurement points



(2) AC test output measurement points



(3) Load conditions



CAUTION

If the load capacitance exceeds 30 pF due to the circuit configuration, it is recommended to insert a buffer in order to reduce the capacitance to a value less than or equal to 30 pF.

37.2 Absolute Maximum Ratings

The absolute maximum ratings are shown in the table below.

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|----------------------------|---------|-----------|------|------|-----------|------|
| Power voltage | VCC | | -0.3 | | 6.5 | V |
| | EnVCC | | -0.3 | | 6.5 | V |
| | A0VCC | | -0.3 | | 6.5 | V |
| | A1VCC | | -0.3 | | 6.5 | V |
| | VDD | | -0.3 | | 1.8 | V |
| Input voltage | VI | EnVCC pin | -0.3 | | EnVCC+0.3 | V |
| | | VCC pin | -0.3 | | VCC+0.3 | V |
| Analogue reference voltage | A0VREFH | | -0.3 | | A0VCC+0.3 | V |
| | A1VREFH | | -0.3 | | A1VCC+0.3 | V |
| Analog input voltage | VAIN | A0VCC pin | -0.3 | | A0VCC+0.3 | V |
| | | A1VCC pin | -0.3 | | A1VCC+0.3 | V |
| Low level output current | IOL | 1 pin | | | 10 | mA |
| | | Total | | | 80 | mA |
| High level output current | IOH | 1 pin | | | -10 | mA |
| | | Total | | | -80 | mA |
| Junction temperature | Tj | | -40 | | 150 | °C |
| Storage temperature | Tstg | | -55 | | 150 | °C |

Note: Ta = 25°C
VSS = EnVSS = A0VSS = A1VSS = 0 V

CAUTIONS

- Using this device without observing these absolute maximum ratings may result in permanent breakdown of the device.
- This device is used in combination of multiple power voltages simultaneously in some cases. Use this device by observing the power pin connections, conditions for combination of power voltages to be applied, voltages that can be applied to pins, and output voltage conditions, which are specified in the manual. Using this device with other than specified power connection or voltage may result in permanent breakdown of the device or damage to the system that contains the device.
- The input voltage, analog reference voltage and analog input voltage must not exceed 6.5 V.

37.3 Supply Voltage Characteristics

| Symbol | MIN. | TYP. | MAX. | Unit |
|----------------------------------|------|------|------|------|
| VCC ^{*1} | 3.0 | | 5.5 | V |
| EnVCC ^{*1} | 3.0 | | 5.5 | V |
| VDD ^{*2} | 1.20 | | 1.35 | V |
| A0VCC A1VCC | 3.0 | | 5.5 | V |
| A0VREFH ^{*3} A1VREFH | 3.0 | | 5.5 | V |

Note: VSS = EnVSS = A0VSS = A1VSS = 0 V

Note 1. VCC and EnVCC should be supplied from the same power source.

Note 2. Applicable only for DPS products.

Note 3. Do not exceed A0VCC or A1VCC.

CAUTION

During operations, supply the specified voltages to all power lines. When stopping operation, turn off all of the power supplies.

37.4 Oscillator Characteristics

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|-------------------|-------------------|-----------|------|------|------|------|
| MainOSC frequency | f _{MOSC} | | | 16 | | MHz |

CAUTION

Oscillation stabilization times differ according to matching with the crystal oscillator. Secure an oscillation stabilization time determined through evaluation of matching.

37.5 Characteristics of High-Speed Internal Oscillator Circuit

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--|----------|-----------|-------|------|-------|------|
| High-speed internal oscillator circuit (HS IntOSC) frequency | f_{RH} | | 15.44 | 16 | 16.56 | MHz |

37.6 PLL Characteristics

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--------------------------------|--------------|-----------|------|------|------|------|
| Input frequency | f_{PLLCLK} | | | 16 | | MHz |
| Output frequency (PLL for CPU) | f_{CPLL} | — | | 160 | | MHz |

37.7 Regulator Characteristics

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|----------------|--------|----------------------|------|------|------|---------|
| Input voltage | VCC | | 3.0 | | 5.5 | V |
| Output voltage | VCL | | 1.15 | 1.25 | 1.35 | V |
| Capacitance | CVCL | Per pin ¹ | 0.07 | 0.1 | 0.13 | μ F |

Note 1. For the connection of the VCL pin, see **Section 2.5, DNF.**

37.8 DC Characteristics

37.8.1 Relationship between Power Name and Pin

The table below shows the relationship between power supply names and pins.

| Pin Name | Power Supply Name | I/O | Input Buffer Type | Note |
|---|---------------------|-----|-------------------|-----------------------------------|
| JP0_4 | VCC | I | SchmittA | 5 V tolerant |
| JP0_2 | E1VCC | I | CMOS/SchmittA | |
| JPx_x other than the above | E1VCC | I/O | CMOS/SchmittB | Output driving ability selectable |
| P0_10 | E1VCC | I/O | CMOS | Output driving ability selectable |
| Py_y Pins listed in Table 2.34, Selection of Input Buffer Characteristics | EnVCC* ¹ | I/O | CMOS/SchmittB | Output driving ability selectable |
| Pz_z other than the above | EnVCC* ¹ | I/O | SchmittB | Output driving ability selectable |
| ADCG0Ixx | A0VCC | I | Analog (ADC) | |
| ADCG1Ixx | A1VCC | I | Analog (ADC) | |
| $\overline{\text{RESET}}$ | VCC | I | SchmittA | 5 V tolerant |
| FLMD0 | E1VCC | I | SchmittC | 5 V tolerant |
| $\overline{\text{ERROROUT}}$ | E0VCC | O | — | |
| $\overline{\text{CVMOUT}}$ | E1VCC | O | — | |
| X1 | VCC | I | — | |
| X2 | VCC | O | — | |

Note 1. Regarding the power supply names (E0VCC or E1VCC), please refer to **Table 2.1, Pin Assignment**.

37.8.2 Buffer Characteristics

Conditions: See Section 37.1.1.1, Common Conditions.

| Item | Symbol | Measuring Condition | MIN. | TYP. | MAX. | Unit | | | |
|---------------------------|--|---------------------------------------|---|--|--|--|----|----------------------------|---|
| High level input voltage | VIH | CMOS | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | $0.65 \times \text{EnVCC}$ | | $\text{EnVCC} + 0.3$ | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | $0.65 \times \text{EnVCC}$ | | $\text{EnVCC} + 0.3$ | V | | |
| | | SchmittA | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | $0.8 \times \text{EnVCC}$ $0.8 \times \text{VCC}$ | | $\text{EnVCC} + 0.3$ $\text{VCC} + 0.3$ | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | $0.75 \times \text{EnVCC}$ $0.75 \times \text{VCC}$ | | $\text{EnVCC} + 0.3$ $\text{VCC} + 0.3$ | V | | |
| | | SchmittB | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | $0.75 \times \text{EnVCC}$ | | $\text{EnVCC} + 0.3$ | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | $0.75 \times \text{EnVCC}$ | | $\text{EnVCC} + 0.3$ | V | | |
| | | SchmittC | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | $0.75 \times \text{EnVCC}$ | | $\text{EnVCC} + 0.3$ | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | $0.7 \times \text{EnVCC}$ | | $\text{EnVCC} + 0.3$ | V | | |
| | | Low level input voltage | VIL | CMOS | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | -0.3 | | $0.35 \times \text{EnVCC}$ | V |
| | | | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | -0.3 | | $0.35 \times \text{EnVCC}$ | V |
| SchmittA | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | | | -0.3 | | $0.25 \times \text{EnVCC}$ $0.25 \times \text{VCC}$ | V | | |
| | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | | | -0.3 | | $0.25 \times \text{EnVCC}$ $0.25 \times \text{VCC}$ | V | | |
| SchmittB | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | | | -0.3 | | $0.4 \times \text{EnVCC}$ | V | | |
| | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | | | -0.3 | | $0.42 \times \text{EnVCC}$ | V | | |
| SchmittC | $3.0\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | | | -0.3 | | $0.3 \times \text{EnVCC}$ | V | | |
| | | | | | | | | | |
| High level output voltage | VOH | | | IOH = -3mA ^{*1} | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | $\text{EnVCC} - 1.0$ | | | V |
| | | | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | $\text{EnVCC} - 1.0$ | | | V |
| | | IOH = -1mA ^{*1} | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | $\text{EnVCC} - 1.0$ | | | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | $\text{EnVCC} - 1.0$ | | | V | | |
| | | IOH = -0.1mA ^{*1} | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | $\text{EnVCC} - 0.5$ | | | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | $\text{EnVCC} - 0.5$ | | | V | | |
| Low level output voltage | VOL | IOL = 3mA ^{*1} | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | | 0.6 | | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | | 0.5 | | V | | |
| | | IOL = 1mA ^{*1} | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | | 0.4 | | V | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | | 0.4 | | V | | |
| Pull-up resistance | RU | Other than below VIN = EnVSS | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | 30 | 60 | 150 | kΩ | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | 20 | 40 | 100 | kΩ | | |
| | | FLMD0 VIN = EnVSS | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | 10 | 19 | 50 | kΩ | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | 10 | 19 | 50 | kΩ | | |
| Pull-down resistance | RD | Other than below VIN = EnVCC | $3.0\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | 15 | 40 | 125 | kΩ | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | 15 | 40 | 100 | kΩ | | |
| | | RESET, FLMD0 VIN = EnVCC | $3.0\text{ V} \leq \text{EnVCC} < 3.6\text{ V}$ | 20 | 60 | 150 | kΩ | | |
| | | | $3.6\text{ V} \leq \text{EnVCC} < 4.3\text{ V}$ | 20 | 60 | 150 | kΩ | | |
| | | | $4.3\text{ V} \leq \text{EnVCC} \leq 5.5\text{ V}$ | 20 | 40 | 100 | kΩ | | |
| | | | | | | | | | |
| Input leak current | ILIH1 | Other than below | Vin = VCC, EnVCC | | | 2 | μA | | |
| | | ADCGnIm (n = 0, 1, m = 0 to 11) | Vin = A0VCC, A1VCC When AD conversion is not performed: | | | 0.2 | μA | | |
| | ILIL1 | Other than below | Vin = 0 V | | | -2 | μA | | |
| | | ADCGnIm (n = 0, 1, m = 0 to 11) | Vin = 0 V When AD conversion is not performed: | | | -0.2 | μA | | |
| Hysteresis width | VH | SchmittA | | $0.27 \times \text{EnVCC}$ | | | V | | |
| | | SchmittB | | $0.1 \times \text{EnVCC}$ | | | V | | |
| | | SchmittC | | $0.2 \times \text{EnVCC}$ | | | V | | |

Note 1. Total load current is 48 mA for the pins when everything is switched on at the same time. However, the total load current must be no greater than 12 mA for pins 38 to 53 on 100-pin devices and no greater than 30 mA for pins 52 to 82 on 144-pin devices.

37.8.3 Allowable Output Current

Conditions: See **Section 37.1.1.1, Common Conditions**.

| Item | Symbol | MIN. | TYP. | MAX. | Unit |
|---|-----------------|------|------|------|------|
| Output high-level allowable current (per pin) | I_{OH} | | | -3 | mA |
| Output high-level allowable current (total)*1 | ΣI_{OH} | | | -48 | mA |
| Output low-level allowable current (per pin) | I_{OL} | | | 3 | mA |
| Output low-level allowable current (total)*1 | ΣI_{OL} | | | 48 | mA |

Note 1. Total load current is 48 mA for the pins when everything is switched on at the same time. However, the total load current must be no greater than 12 mA for pins 38 to 53 on 100-pin devices and no greater than 30 mA for pins 52 to 82 on 144-pin devices.

CAUTION

This item affects the calorific value and T_j of the chip. In addition to these restrictions, take thermal design into consideration.

37.8.4 Injection Current

Conditions: See **Section 37.1.1.1, Common Conditions**.

| Item | | Symbol | MIN. | TYP. | MAX. | Unit |
|--------------------------------|--------------|----------|------|------|------|------|
| DC injection current (per pin) | Digital pin | IINJ_DIN | -2 | | 2 | mA |
| | Analogue pin | IINJ_AIN | -3 | | 3 | mA |
| DC injection current (Total) | | IINJ_TOT | -50 | | 50 | mA |

37.8.5 Input Capacitance

Conditions: See **Section 37.1.1.1, Common Conditions**.

As for temperature conditions, refer measurement conditions.

| Item | Symbol | MIN. | TYP. | MAX. | Unit | Measuring Condition |
|--------------------------|--------|------|------|------|------|--|
| Input capacitance | CI | | 10 | | pF | f = 1 MHz |
| Input/output capacitance | CIO | | 10 | | pF | Pins not to be measured: 0 V $T_j = 25^\circ\text{C}$ |
| Output capacitance | CO | | 10 | | pF | |
| Input capacitance for X1 | CX1 | | 20 | | pF | |

NOTE

For the analogue input pins (ADCGnIm), see the description of the equal input capacitance in related **Section 37.14, A/D Convertor Characteristics**.

37.8.6 Power Current Characteristics

Table 37.1 Power Current of eVR Product

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | | Symbol | MIN. | TYP. | MAX. | Unit |
|---------------------------------|---|----------|------|------|------|------|
| VCC power current ^{*1} | Run mode | IVCCEVR | | | 230 | mA |
| | When BIST is executed | IVCCBEVR | | | 230 | mA |
| | When rewriting flash memory ^{*2} | IVCCFEVR | | | 230 | mA |
| AnVCC power current | | IAVCC | | | 6.5 | mA |
| AnVREFH current | | IAVREF | | | 0.5 | mA |

Note 1. The MAX. value includes the operating current of peripheral devices. However, the current in A/D converter, I/O port, or on-chip pull-up/-down resistor is not included.

Note 2. This is the value of the power current when rewriting code flash memory or data flash memory.

Table 37.2 Power Current of DPS Product

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | | Symbol | MIN. | TYP. | MAX. | Unit |
|---------------------------------|---|----------|------|------|------|------|
| VDD power current ^{*1} | Run mode | IVDDDPS | | | 210 | mA |
| | When BIST is executed | IVDDBDPS | | | 210 | mA |
| | When rewriting flash memory ^{*2} | IVDDFDPS | | | 185 | mA |
| VCC power current | Run mode | IVCCDPS | | | 20 | mA |
| | When BIST is executed | IVCCBDPS | | | 20 | mA |
| | When rewriting flash memory ^{*2} | IVCCFDPS | | | 45 | mA |

Note 1. The MAX. value includes the operating current of peripheral devices. However, the current in A/D converter, I/O port, or on-chip pull-up/-down resistor is not included.

Note 2. This is the value of the power current when rewriting code flash memory or data flash memory.

Note 3. AnVCC and AnVREFH current are the same as those of the eVR products.

37.9 AC Characteristics

37.9.1 Power UP/Down Timing

Table 37.3 eVR Products

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | MIN. | MAX. | Unit | Remarks |
|---|-------------|------|------|---------|---------|
| Pin reset time at low level for power on | t_{RESW1} | 10 | | ms | *1 |
| Pin reset time at low level for power off | t_{RESW2} | 2 | | μ s | *2 |
| Operating mode setup time | t_{MDS} | 1 | | ms | |
| Operating mode hold time | t_{MDH} | 1 | | ms | |
| VCC/AnVCC voltage ramp | t_{VS1} | 0.02 | 500 | V/ms | |

- Note 1. t_{RESW1} is the time over which assertion of the reset signal is required to continue until the supply of internal clock signals becomes stable after all power supplies have been turned on. Release the pin reset after the periods of t_{RESW1} and t_{OSC} have elapsed. VCC and EnVCC should be supplied from the same source. No restriction applies to potential differences between the levels of the VCC/EnVCC supplies and the AnVCC supply during power up.
- Note 2. t_{RESW2} is the time over which assertion of the reset signal is required until either of the power voltages has dropped below the lower limit.
- Note 3. The states of I/O pins are not reset during the noise cancellation interval (max. 1.2 μ s) of the reset signal following its assertion while power is being turned off. During that time, do not allow any input of mid-range potential to a pin or the contention of output data.
- Note 4. If the power is turned off during the programming or erasure of flash memory, the data in the area that was programmed or erased cannot be guaranteed.

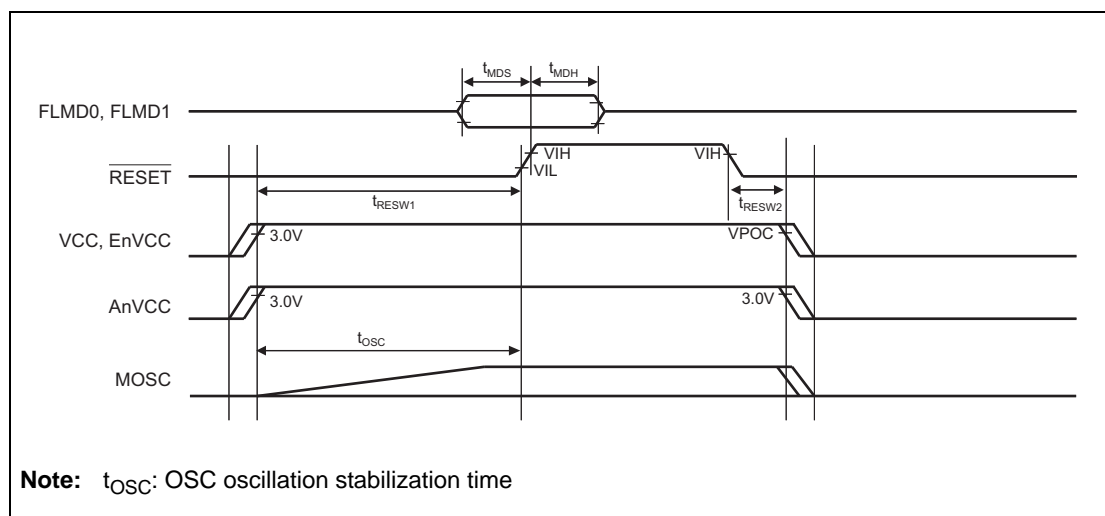


Figure 37.1 Power UP/Down Timing: eVR Products

Table 37.4 DPS Products

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | MIN. | MAX. | Unit | Remarks |
|---|-------------|------|------|---------|---------|
| Pin reset time at low level for power on | t_{RESW1} | 10 | | ms | *1 |
| Pin reset time at low level for power off | t_{RESW2} | 2 | | μ s | *2 |
| Operating mode setup time | t_{MDS} | 1 | | ms | |
| Operating mode hold time | t_{MDH} | 1 | | ms | |
| VCC/AnVCC voltage ramp | t_{VS1} | 0.02 | 500 | V/ms | |
| VDD voltage ramp | t_{VS2} | 2 | 500 | V/ms | |

- Note 1. t_{RESW1} is the time over which assertion of the reset signal is required to continue until the supply of internal clock signals becomes stable after all power supplies have been turned on. Release the pin reset after the periods of t_{RESW1} and t_{OSC} have elapsed. VCC and EnVCC should be supplied from the same source. No restriction applies to potential differences between the levels of the VCC/EnVCC supplies, the AnVCC supply, and the VDD supply during power up.
- Note 2. t_{RESW2} is the time over which assertion of the reset signal is required until either of the power voltages has dropped below the lower limit.
- Note 3. The states of I/O pins are not reset during the noise cancellation interval (max. 1.2 μ s) of the reset signal following its assertion while power is being turned off. During that time, do not allow any input of mid-range potential to a pin or the contention of output data.
- Note 4. If the power is turned off during the programming or erasure of flash memory, the data in the area that was programmed or erased cannot be guaranteed.

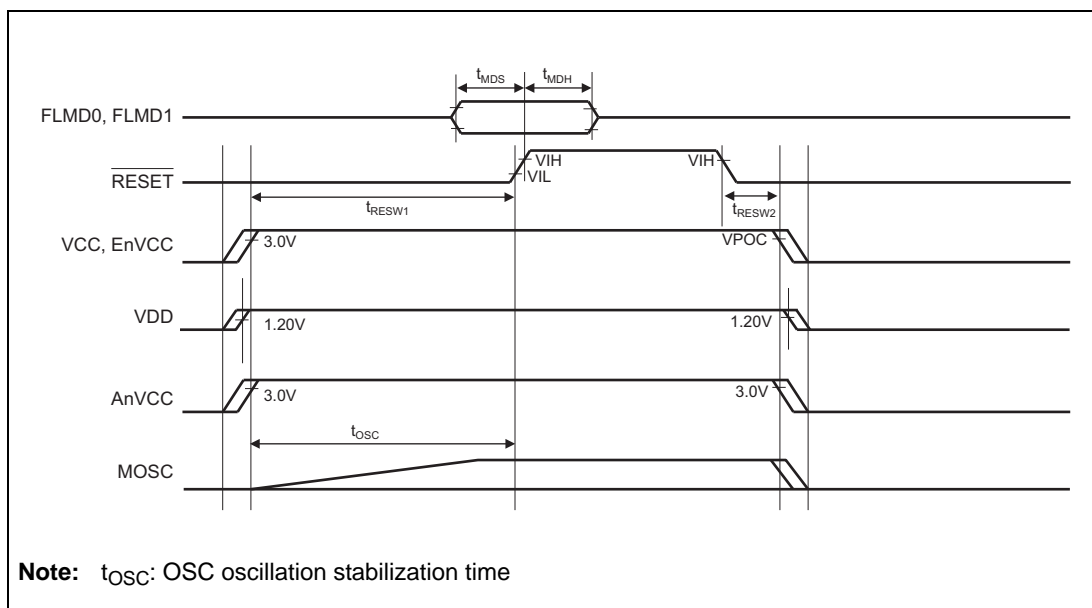


Figure 37.2 Power UP/Down Timing: DPS Products

37.9.2 Driving Ability

Conditions: See Section 37.1.1.1, Common Conditions.

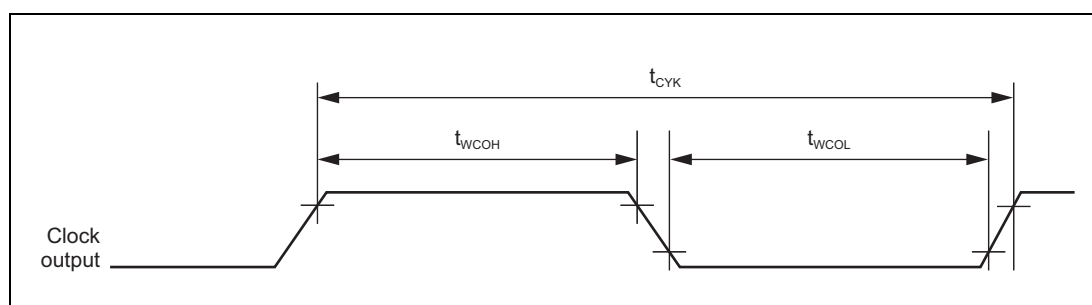
| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|----------------------------|--------|-------------------|------|------|------|------|
| Output frequency | | High-speed mode | | | 20 | MHz |
| | | Medium-speed mode | | | 14 | MHz |
| | | Low-speed mode | | | 4 | MHz |
| Output rising/falling time | | High-speed mode | | | 10 | ns |
| | | Medium-speed mode | | | 20 | ns |
| | | Low-speed mode | | | 50 | ns |

37.9.3 Clock Output Timing

Conditions: See Section 37.1.1.1, Common Conditions.

Output pin are the fast mode.

| Item | Symbol | MIN. | TYP. | MAX. | Unit |
|-------------------------------|------------|------------------|------|------|------|
| Clock output cycle time | t_{CYK} | 50 (20MHz) | | | ns |
| Clock output high level width | t_{WCOH} | $t_{CYK}/2 - 12$ | | | ns |
| Clock output low level width | t_{WCOL} | $t_{CYK}/2 - 12$ | | | ns |

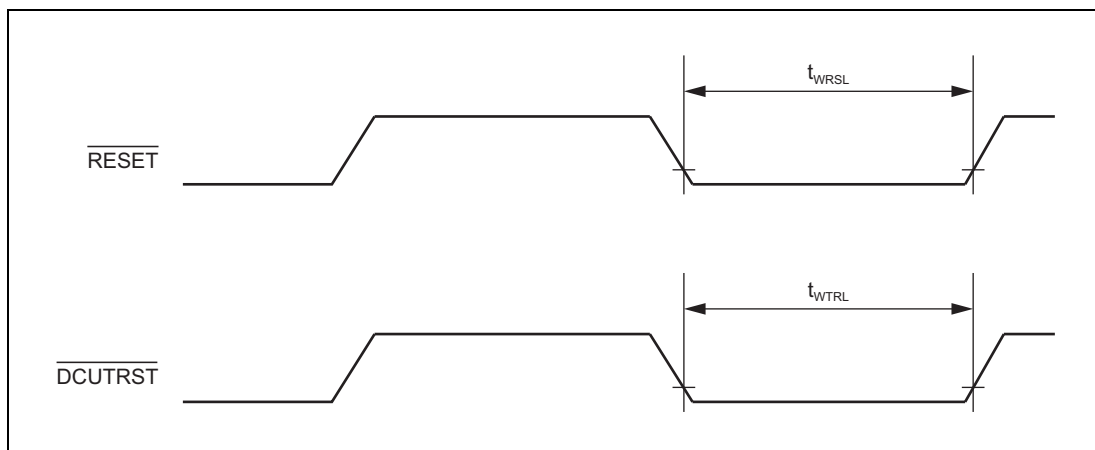


37.9.4 Control Signal Timing

37.9.4.1 Reset

Conditions: See Section 37.1.1.1, Common Conditions.

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|-------------------------------|------------|---------------------|------|------|------|---------------|
| RESET input low level width | t_{WRSL} | Other than power-on | 1.5 | — | — | μs |
| DCUTRST input low level width | t_{WTRL} | | 1.5 | — | — | μs |



37.9.4.2 Interrupts/ADTRG

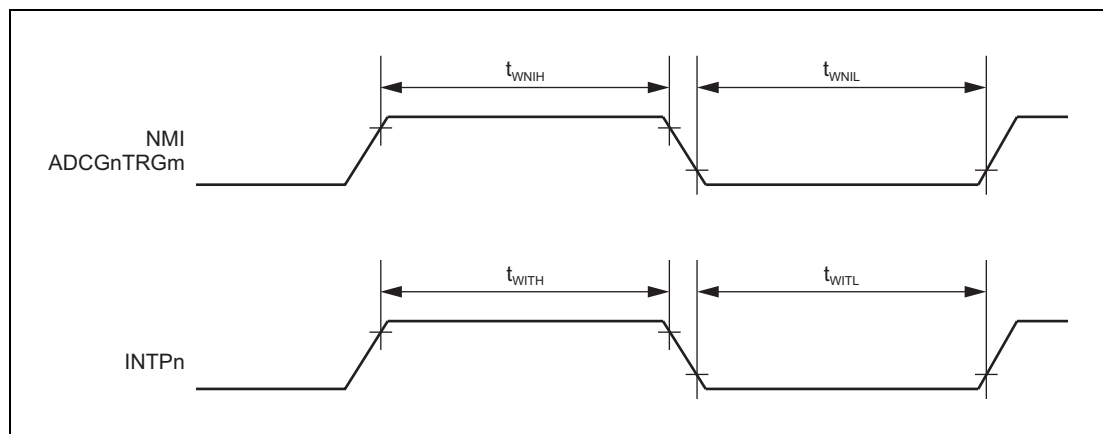
Conditions: See Section 37.1.1.1, Common Conditions.

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|----------------------------------|------------|---------------------------------------|--------------------------------------|------|------|------|
| NMI input high level width | t_{WNIH} | Digital filter | $k \times T_{\text{samp}} + 20^{*1}$ | | | ns |
| NMI input low level width | t_{WNIL} | Digital filter | $k \times T_{\text{samp}} + 20^{*1}$ | | | ns |
| INTPn input high level width | t_{WITL} | Digital filter | $k \times T_{\text{samp}} + 20^{*1}$ | | | ns |
| | | Analog filter, n = 7, 8 | 600 | | | ns |
| INTPn input low level width | t_{WITL} | Digital filter | $k \times T_{\text{samp}} + 20^{*1}$ | | | ns |
| | | Analog filter, n = 7, 8 | 600 | | | ns |
| ADCGnTRGm input high level width | t_{WNIH} | Digital filter, n = 0,1, m = 0, 1 | $k \times T_{\text{samp}} + 20^{*1}$ | | | ns |
| ADCGnTRGm input low level | t_{WNIL} | Digital filter, n = 0, 1, m = 0, 1 | $k \times T_{\text{samp}} + 20^{*1}$ | | | ns |

Note 1. k is the number of samples taken by the digital noise filter for the input.
 T_{samp} is the sampling interval of the digital noise filter for the input.

CAUTION

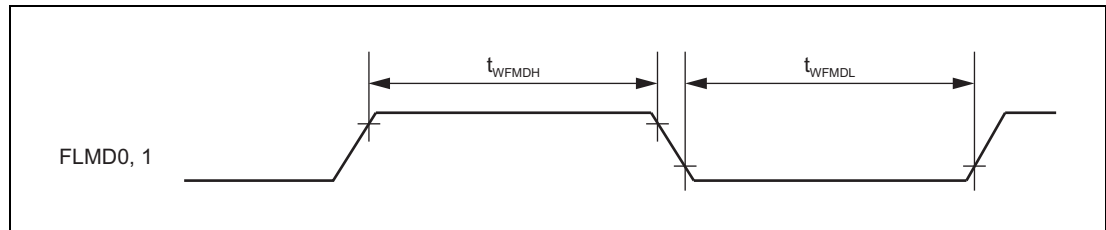
The specifications above indicate the pulse width to be always detected as an effective edge. Even when a pulse width smaller than above is input, such value may be detected as an effective edge.



37.9.4.3 Mode

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--------------------------------|-------------|-----------|------|------|------|------|
| FLMD0,1 input high level width | t_{WFMDH} | | 600 | | | ns |
| FLMD0,1 input low level width | t_{WFMDL} | | 600 | | | ns |



37.9.5 Timer Timing

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---|-------------------------------|-----------|----------------------------|------|------|------|
| TAUDnIm input high/low level width (n = 3, m = 0 to 15) | $t_{WTDIH}/$ t_{WTDIL} | | $k \times Tsamp + 20^{*1}$ | | | ns |
| TAUJnIm input high/low level width (n = 3, m = 0 to 3) | $t_{WTJIH}/$ t_{WTJIL} | | $k \times Tsamp + 20^{*1}$ | | | ns |
| ENCAnTINm input high/low level width (n = 0, 1; m = 0, 1) | $t_{WTDIH}/$ t_{WTDIL} | | $k \times Tsamp + 20^{*1}$ | | | ns |
| ENCAnEm input high/low level width (n = 0, 1, m = 0, 1, C) | $t_{WENIH}/$ t_{WENIL} | | $k \times Tsamp + 20^{*1}$ | | | ns |
| TSG3nPTSIIm input high/low level width (n = 0, 1, m = 0, 1, 2) | $t_{WTSPIH}/$ t_{WTSPIL} | | $k \times Tsamp + 20^{*1}$ | | | ns |
| TSG3nCLKI input high/low level width (n = 0, 1) | $t_{WTSCIH}/$ t_{WTSCIL} | | $k \times Tsamp + 20^{*1}$ | | | ns |
| TAPAnESO input high/low level width (n = 0, 1) | $t_{WESIH}/$ t_{WESIL} | | 600 | | | ns |

Note 1. k is the number of samples taken by the digital noise filter for the input.
Tsamp is the sampling interval of the digital noise filter for the input.

37.9.6 CSI Timing

37.9.6.1 CSIG Timing

Conditions: See **Section 37.1.1.1, Common Conditions**.
Output are specified for buffer "fast mode".

Table 37.5 CSIG Timing (Master Mode)

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--|--------------|--|---------------------------|------|------|------|
| Macro operation clock cycle time | t_{KCYGn} | | 12.5 (max. 80 MHz) | | | ns |
| CSIGnSC cycle time | t_{KCYMGn} | | 125 | | | ns |
| CSIGnSC high level width | t_{KWHMGn} | | $t_{KCYMGn} / 2 - 12$ | | | ns |
| CSIGnSC low level width | t_{KWLMGn} | | $t_{KCYMGn} / 2 - 12$ | | | ns |
| CSIGnSI setup time (vs. CSIGnSC) | t_{SSIMGn} | | 30 | | | ns |
| CSIGnSI hold time (vs. CSIGnSC) | t_{HSIMGn} | | 0 | | | ns |
| CSIGnSO output delay time (vs. CSIGnSC) | t_{DSOMGn} | | | | 7 | ns |
| CSIGnRYI setup time (vs. CSIGnSC) | t_{SRYIGn} | CSIGnCTL1.CSIGnSIT = x CSIGnCTL1.CSIGnHSE = 1 | $2 \times t_{KCYGn} + 25$ | | | ns |

Table 37.6 CSIG Timing (Slave Mode)

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---|--------------|-----------|-----------------------|------|------|------|
| Macro operation clock cycle time | t_{KCYGn} | | 12.5 (max. 80 MHz) | | | ns |
| CSIGnSC cycle time | t_{KCYSGn} | | 150 | | | ns |
| CSIGnSC high level width | t_{KWHSGn} | | $t_{KCYSGn} / 2 - 12$ | | | ns |
| CSIGnSC low level width | t_{KWLSGn} | | $t_{KCYSGn} / 2 - 12$ | | | ns |
| CSIGnSI setup time (vs. CSIGnSC) | t_{SSISGn} | | 20 | | | ns |
| CSIGnSI hold time (vs. CSIGnSC) | t_{HSISGn} | | $t_{KCYGn} + 5$ | | | ns |
| CSIGnSO output delay time (vs. CSIGnSC) | t_{DSOSGn} | | | | 30 | ns |
| CSIGnRYO output delay time (vs. CSIGnSC) | t_{SRYOGn} | | | | 38 | ns |

37.9.6.2 CSIH Timing

Conditions: See **Section 37.1.1.1, Common Conditions**.
Output are specified for buffer "fast mode".

Table 37.7 CSIH Timing (Master Mode)

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--|----------------|--|--|------|------|------|
| Macro Operation clock cycle time | t_{KCYHn} | | 12.5 (max. 80MHz) | | | ns |
| CSIHnSC cycle time | t_{KCYMHn} | | 100 | | | ns |
| CSIHnSC high level width | t_{KWHMHn} | | $t_{KCYMHn}/2 - 12$ | | | ns |
| CSIHnSC low level width | t_{KWLMHn} | | $t_{KCYMHn}/2 - 12$ | | | ns |
| CSIHnSI setup time (vs. CSIHnSC) | t_{SSIMHn} | | 25 | | | ns |
| CSIHnSI hold time (vs. CSIHnSC) | t_{HSIMHn} | | 0 | | | ns |
| CSIHnSO output delay (vs. CSIHnSC) | t_{DSOMHn} | | | 7 | | ns |
| CSIHnRYI setup time (vs. CSIHnSC) | t_{SRYIHn} | CSIHnCTL1.CSIHnSIT = x CSIHnCTL1.CSIHnHSE = 1 | $2 \times t_{KCYMHn} + 25$ | | | ns |
| CSIHnCSS0 to CSIHnCSS7 inactive level width | $t_{WSCSBHn}$ | | $CSIDLE \times t_{KCYMHn} - 20$ | | | ns |
| CSIHnCSS0 to CSIHnCSS7 setup time (vs. CSIHnSC) | $t_{SSCSBHn0}$ | CSIHnCFGx.CSIHnDAPx = 0 | $CSSETUP \times t_{KCYMHn} - 10$ | | | ns |
| | $t_{SSCSBHn1}$ | CSIHnCFGx.CSIHnDAPx = 1 | $(CSSETUP + 0.5) \times t_{KCYMHn} - 10$ | | | ns |
| CSIHnCSS0 to CSIHnCSS7 hold time (vs. CSIHnSC) | $t_{HSCSBHn0}$ | CSIHnCTL1.CSIHnSIT = 0 | $CSHOLD \times t_{KCYMHn} - 10$ | | | ns |
| | $t_{HSCSBHn1}$ | CSIHnCTL1.CSIHnSIT = 1 | $(CSHOLD + 0.5) \times t_{KCYMHn} - 10$ | | | ns |

Note: CSSETUP: The values set in CSIHnCFGx.CSIHnSPx[3:0]
CSHOLD: The values set in CSIHnCFGx.CSIHnHDx[3:0]
CSIDLE: The value set in CSIHnCFGx.CSIHnIDx[2:0]

Table 37.8 CSIH Timing (Slave Mode)

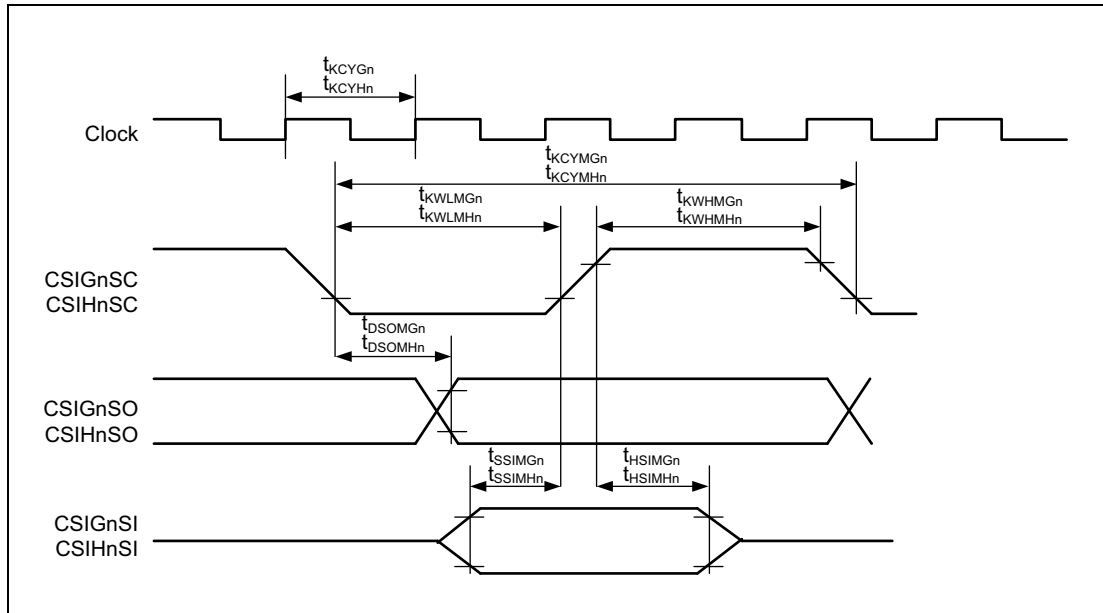
| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---|---------------|-----------|----------------------|------|------|------|
| Macro Operation clock cycle time | t_{KCYHn} | | 12.5 (max. 80MHz) | | | ns |
| CSIHnSC cycle time | t_{KCYSHn} | | 150 | | | ns |
| CSIHnSC high level width | t_{KWHSHn} | | $t_{KCYSHn}/2 - 12$ | | | ns |
| CSIHnSC low level width | t_{KWLSHn} | | $t_{KCYSHn}/2 - 12$ | | | ns |
| CSIHnSI setup time (vs. CSIHnSC) | t_{SSISHn} | | 20 | | | ns |
| CSIHnSI hold time (vs. CSIHnSC) | t_{HSISHn} | | $t_{KCYHn} + 5$ | | | ns |
| CSIHnSO output delay time (vs. CSIHnSC) | t_{DSOSHn} | | | 30 | | ns |
| CSIHnRYO output delay time (vs. CSIHnSC) | t_{SRYOHn} | | | 38 | | ns |
| CSIHnSSI setup time (vs. CSIHnSC) | $t_{SSSISHn}$ | | $t_{KCYSHn}/2 - 5$ | | | ns |
| CSIHnSSI hold time (vs. CSIHnSC) | $t_{HSSISHn}$ | | $t_{KCYHn} + 30$ | | | ns |

37.9.6.3 CSIG/CSIH Timing Charts

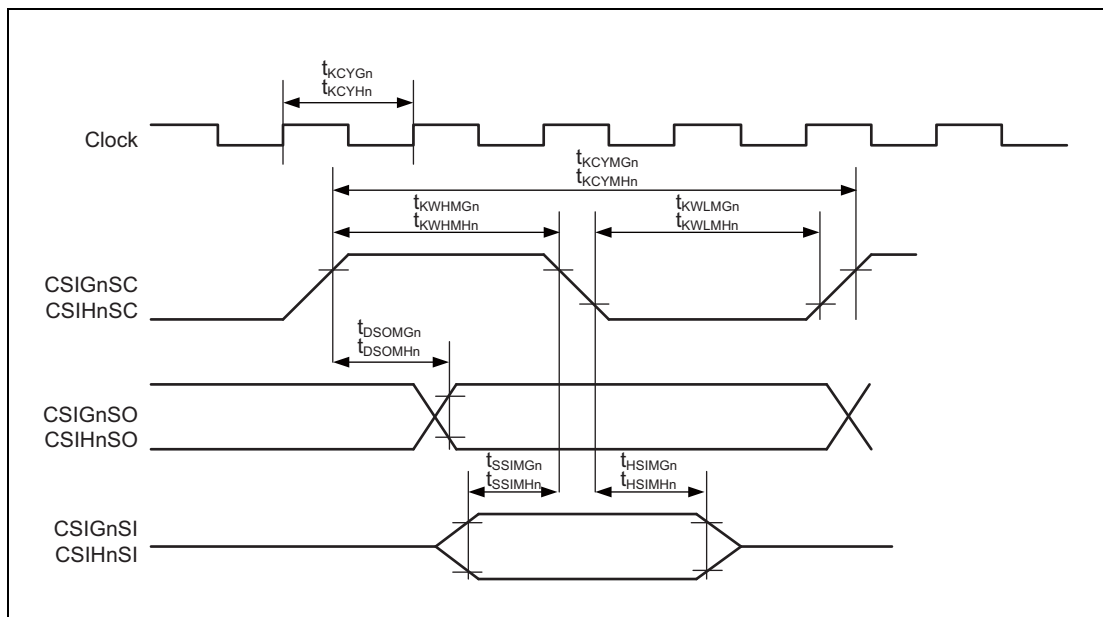
(1) SCKO/SI/SO

Master Mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)

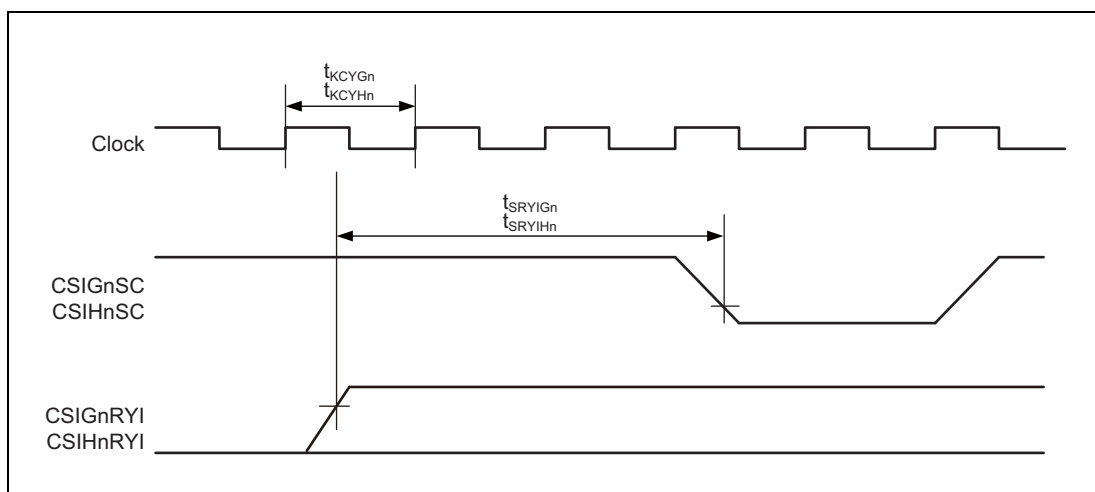


- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

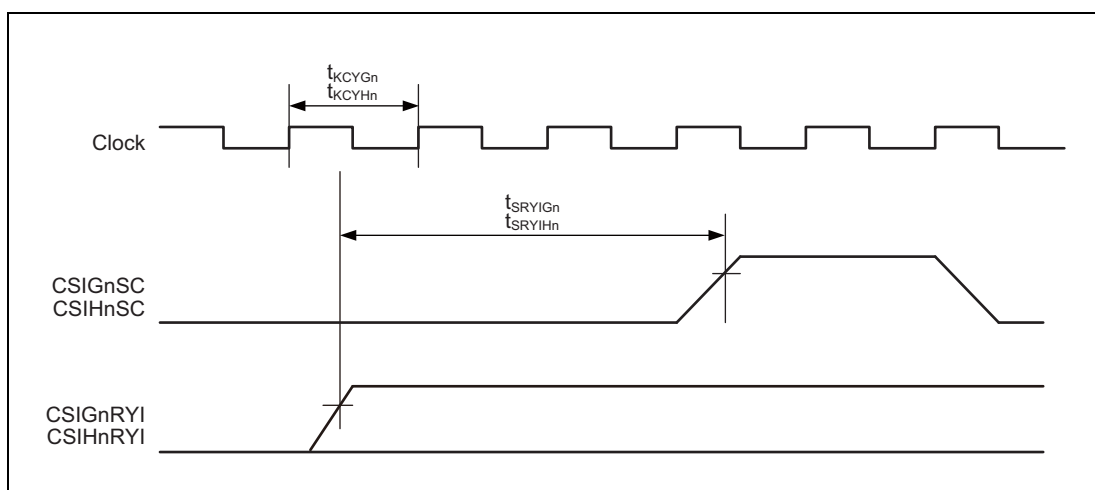


(2) RYI

- CSIG: Only master mode (CSIGnCTL1: CSIGnHSE = 1, CSIGnCTL1: CSIGnSIT = 0)
 - CSIH: Only master mode (CSIHnCTL1: CSIHnHSE = 1, CSIHnCTL1: CSIHnSIT = 0)
- CSIG (CSIGnCTL1: CSIGnCKR = 0)
 - CSIH (CSIHnCFGx: CSIHnCKPx = 0)



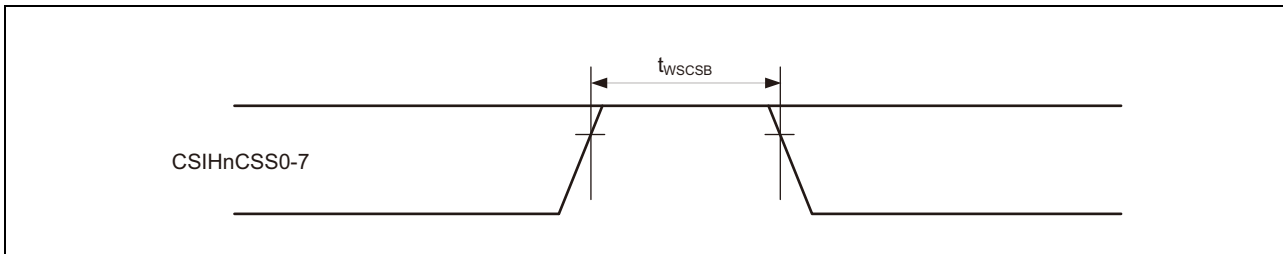
- CSIG (CSIGnCTL1: CSIGnCKR = 1)
- CSIH (CSIHnCFGx: CSIHnCKPx = 1)



(3) CSSn

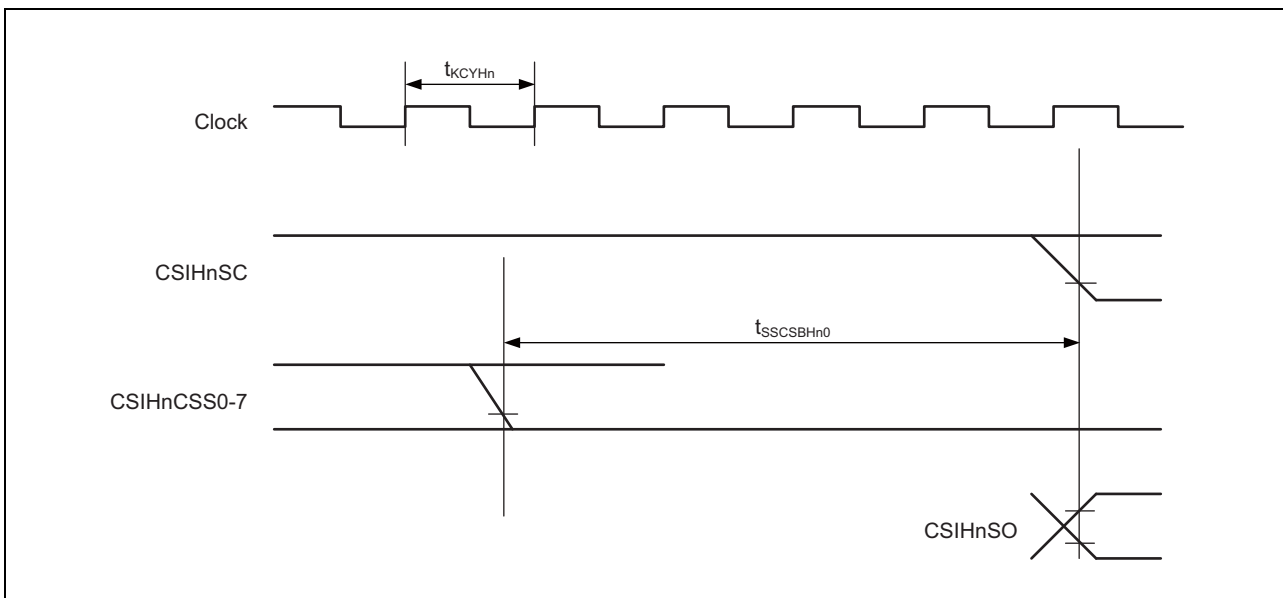
Only Master Mode (Inactive Level Width):

- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0

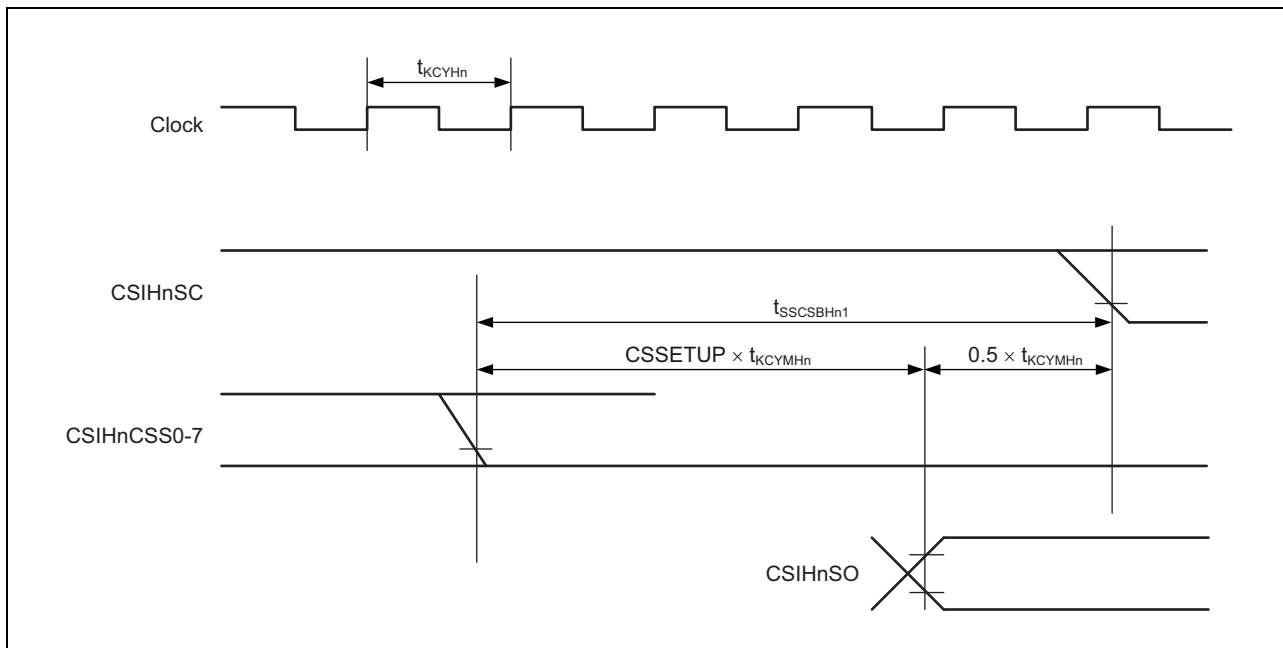


Only Master Mode (Setup Time):

- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0

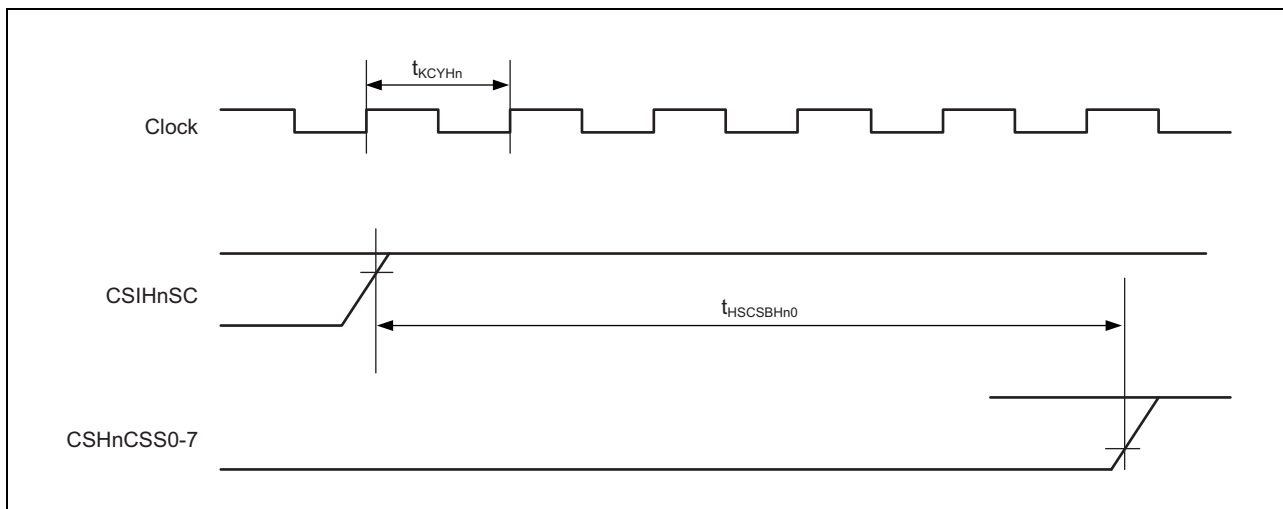


- CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 1

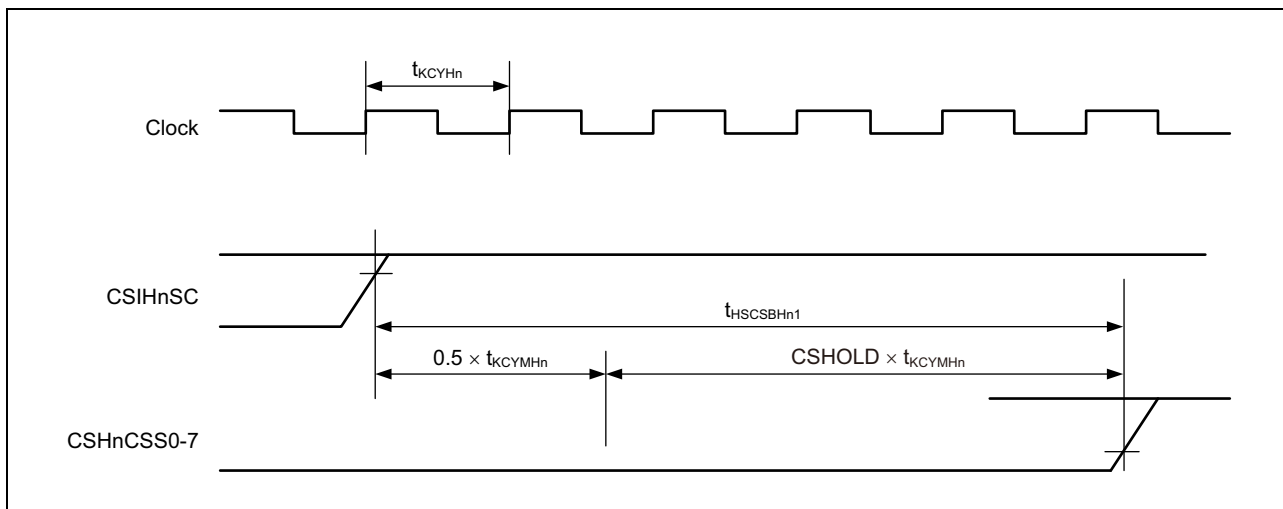


Only Master Mode (Hold Time):

- CSIHnCTL1: CSIHnSIT = 0, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



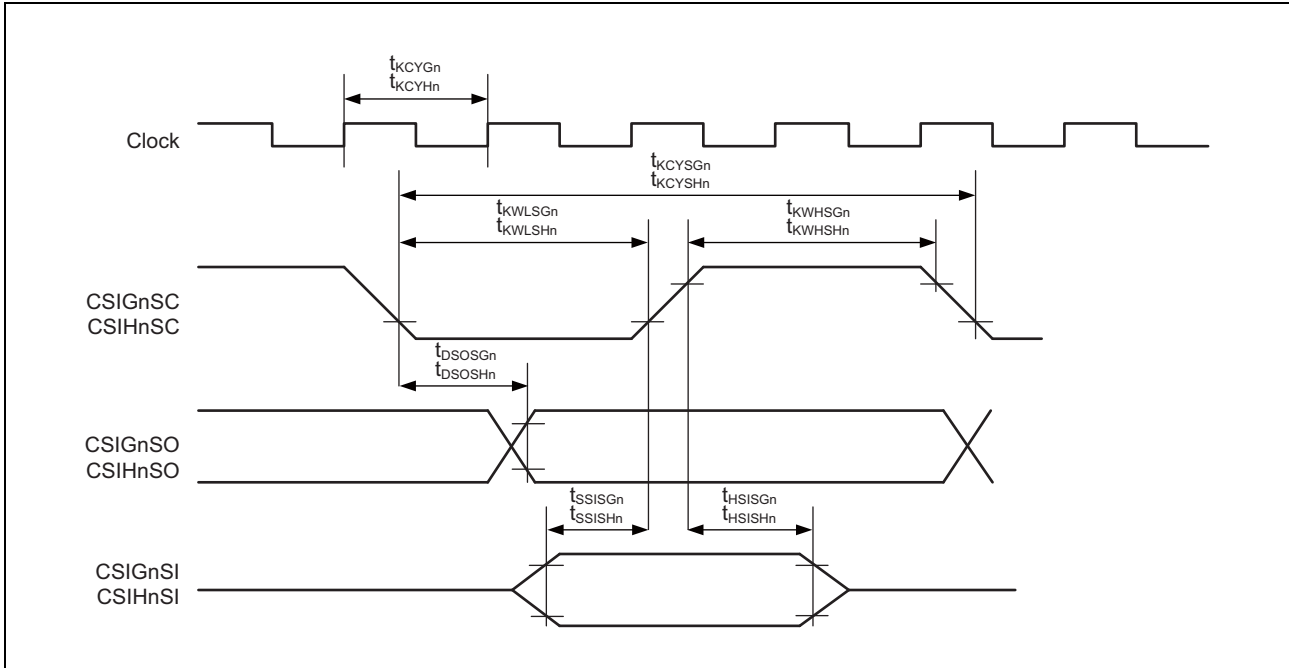
- CSIHnCTL1: CSIHnSIT = 1, CSIHnCFGx: CSIHnCKPx = 0, CSIHnCFGx: CSIHnDAPx = 0



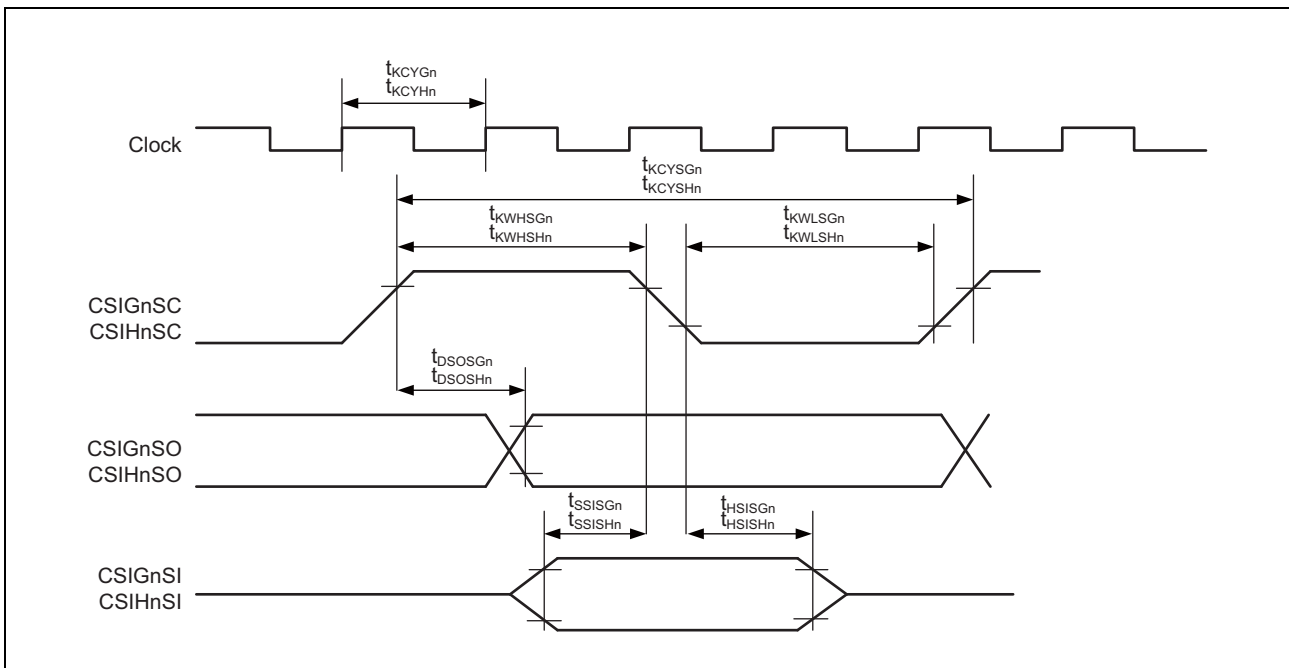
(4) SCKO/SI/SO

Slave Mode:

- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/0 or 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)

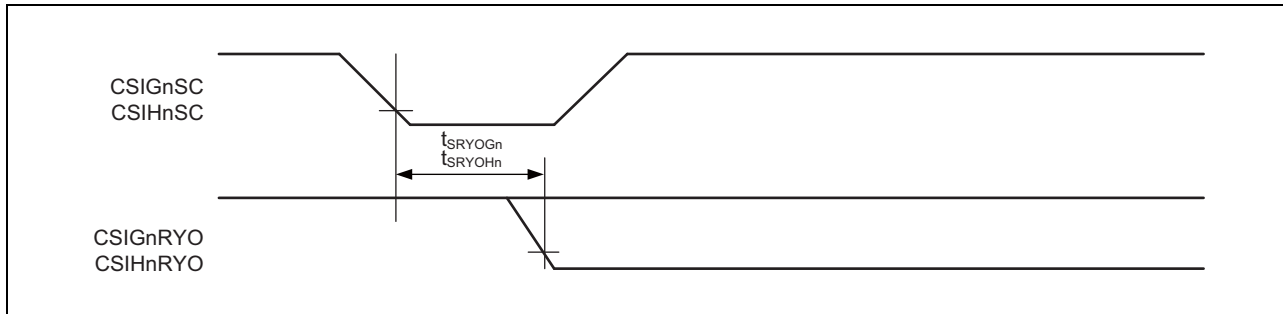


- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0 or 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)

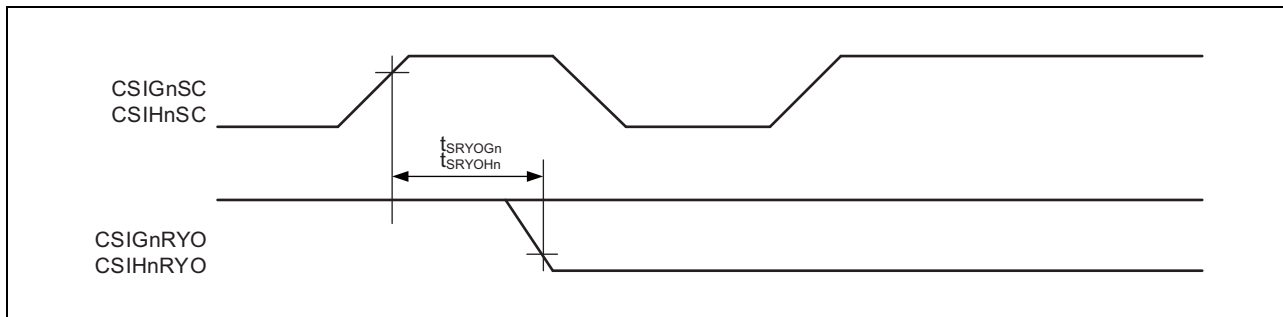


(5) RYO

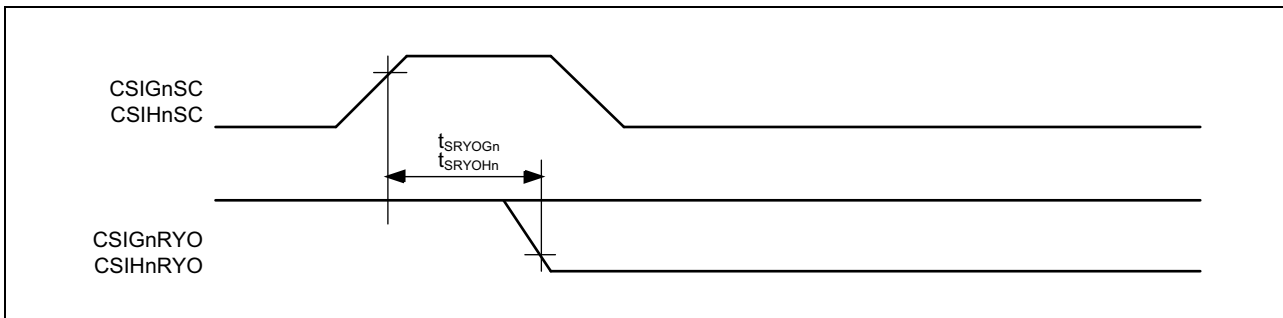
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP0 = 0/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0)



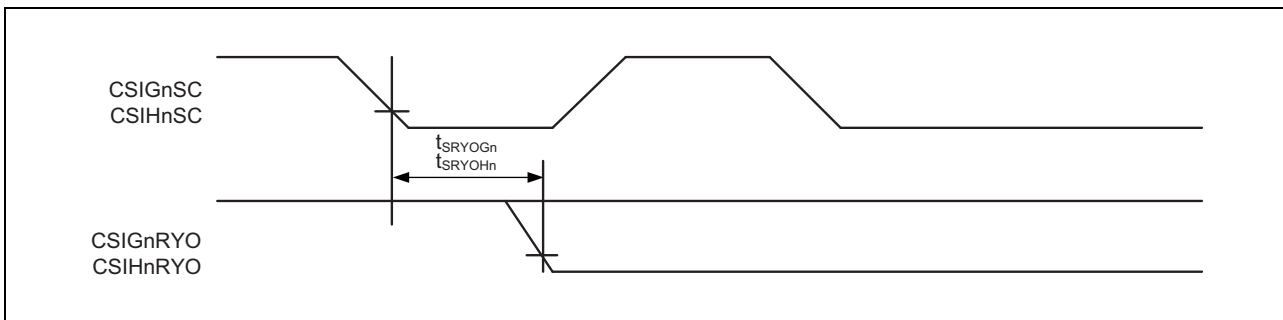
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 0/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/1)



- CCSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/0)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0)



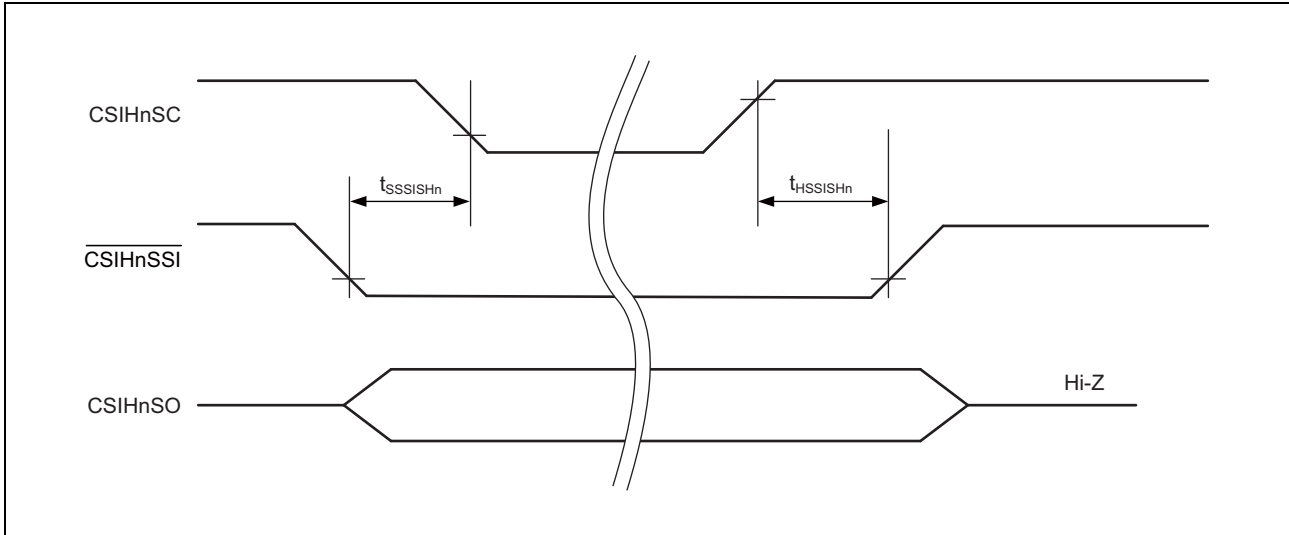
- CSIG (CSIGnCTL1: CSIGnCKR/CSIGnCFG0: CSIGnDAP = 1/1)
- CSIH (CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/1)



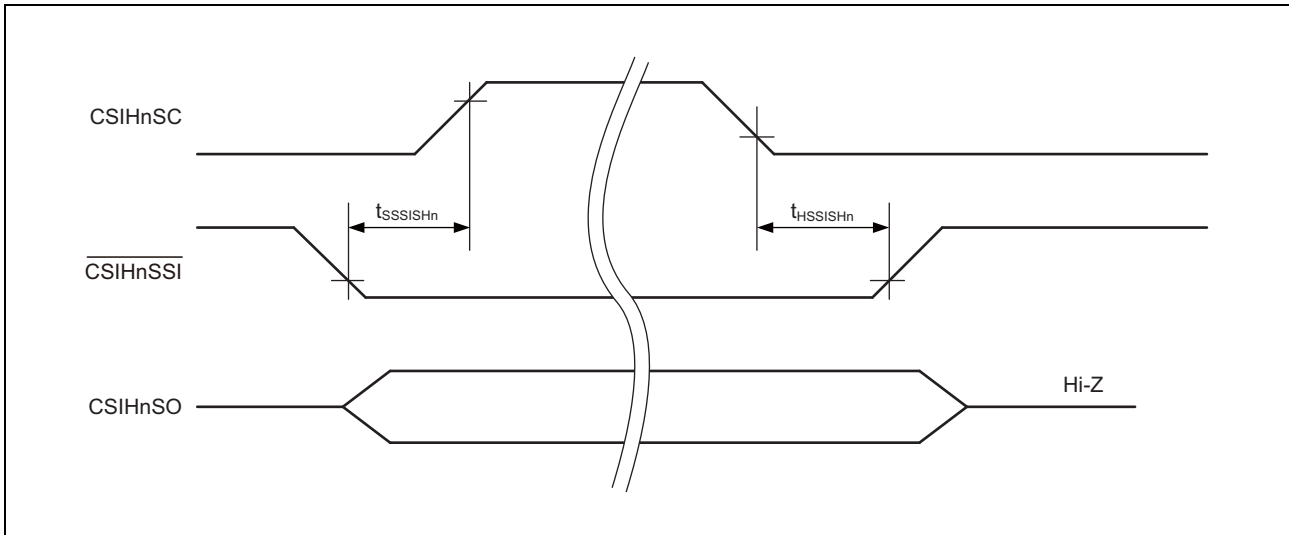
(6) SSI

Slave Mode:

- CSH (CSIHnCTL1: CSIHnSSE=1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 0/0 or 1/1)



- CSH (CSIHnCTL1: CSIHnSSE=1, CSIHnCFGx: CSIHnCKPx/CSIHnCFGx: CSIHnDAPx = 1/0 or 0/1)



37.9.7 SCI3 Timing

Conditions: See **Section 37.1.1.1, Common Conditions**.
 Output pin are specified for buffer "fast mode".

Table 37.9 SCI3 Timing (Master Mode)

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--------------------------|------------|-------------------|----------------------|------|----------------------|------|
| Output cycle time | t_{Syc} | Asynchronous | $16 \times t_{PCLK}$ | | | ns |
| | | Clock synchronous | $16 \times t_{PCLK}$ | | | ns |
| Output clock pulse width | t_{SCKW} | | $0.4 \times t_{Syc}$ | | $0.6 \times t_{Syc}$ | ns |
| Transmit data delay time | t_{TXD} | | | | 40 | ns |
| Receive data setup time | t_{RXS} | Clock synchronous | $2 \times t_{PCLK}$ | | | ns |
| Receive data hold time | t_{RXH} | Clock synchronous | $2 \times t_{PCLK}$ | | | ns |

Note: t_{PCLK} is the operating clock of SCI3.

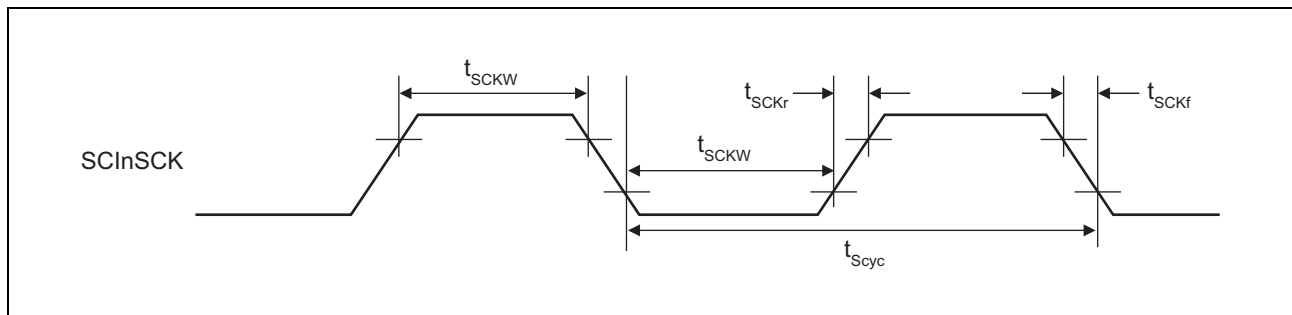


Figure 37.3 SCI Clock Input/Output Timing

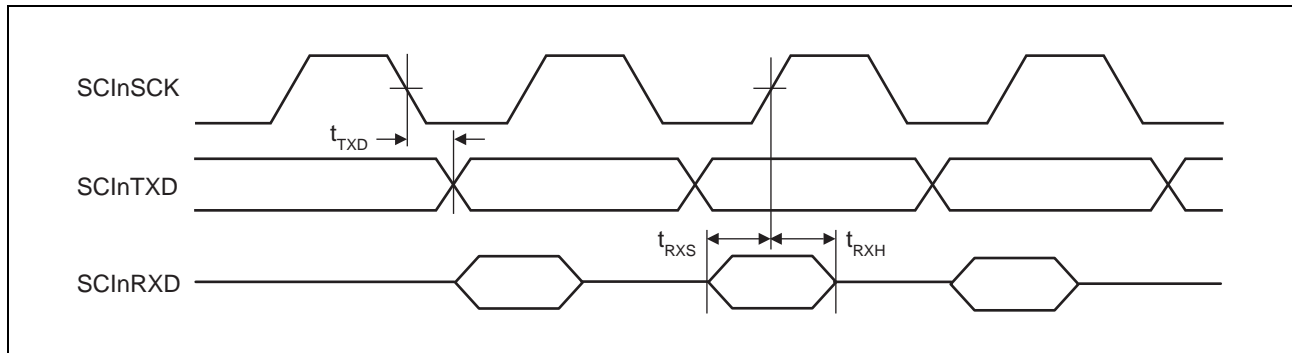


Figure 37.4 SCI Input/Output Timing, Clock Synchronous Mode (in Master Mode)

Table 37.10 SCI3 Timing (Slave Mode)

| Item | Symbol | MIN. | TYP. | MAX. | Unit |
|--|------------|-----------------------|------|--------------------------|------|
| Input cycle time | t_{Scyc} | $12 \times t_{PCLK}$ | | | ns |
| Input clock pulse width | t_{SCKW} | $0.4 \times t_{Scyc}$ | | $0.6 \times t_{Scyc}$ | ns |
| Transmit data delay time ^{*1} | t_{TXD} | $2 \times t_{PCLK}$ | | $50 + 3 \times t_{PCLK}$ | ns |
| Input clock rising time | t_{SCKr} | | 20 | | ns |
| Input clock falling time | t_{SCKf} | | 20 | | ns |
| Receive data setup time | t_{RXS} | $2 \times t_{PCLK}$ | | | ns |
| Receive data hold time | t_{RXH} | $2 \times t_{PCLK}$ | | | ns |

Note 1. This does not apply to transmission of Data 0 (the first bit), which is not transferred in continuous transfer mode.

Transmission of Data 0 (the first bit), which is not transferred in continuous transfer mode, starts when TDRE becomes 0.

Note: t_{PCLK} is the operating clock of SCI3.

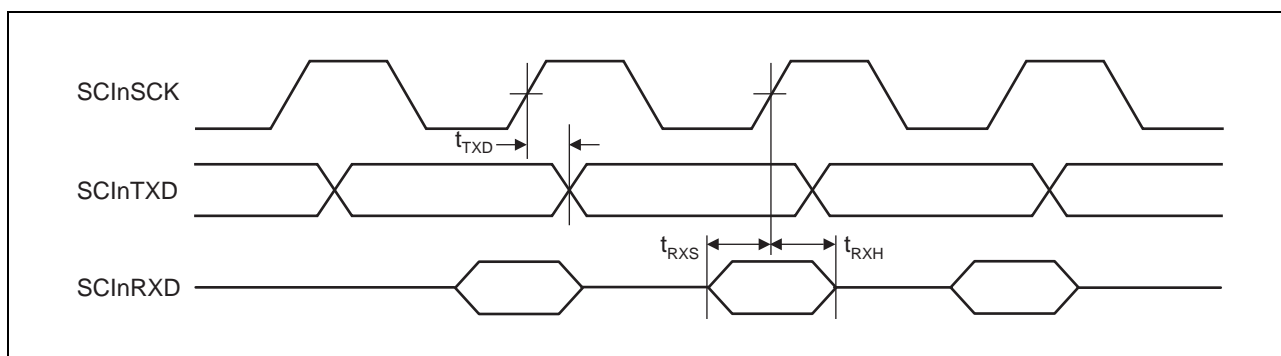


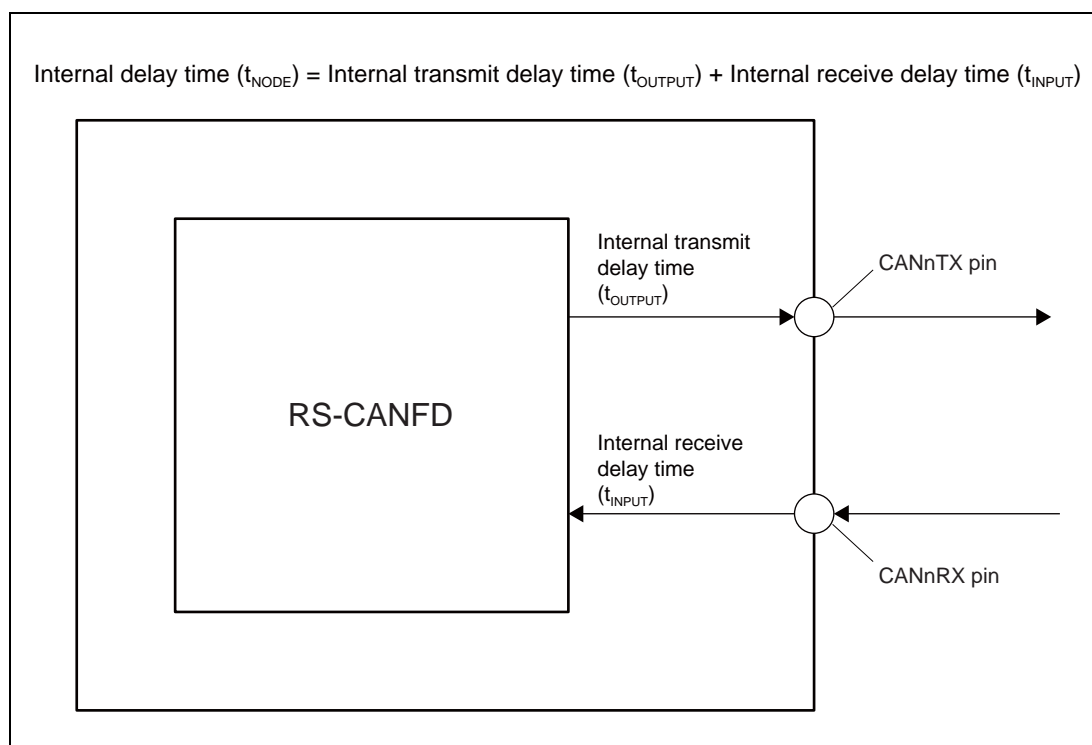
Figure 37.5 SCI Input/Output Timing, Clock Synchronous Mode (in Slave Mode)

37.9.8 RS-CANFD Timing

Conditions: See **Section 37.1.1.1, Common Conditions**.

Output signals are specified for the buffer "fast mode" and "middle mode".

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---------------------|------------|------------------------------------|------|------|------|------|
| Internal delay time | t_{NODE} | | | | 50 | ns |
| Transfer rate | | CAN FD arbitration and CAN 2.0B | | | 1 | Mbps |
| | | CAN FD data phase | | | 8 | Mbps |



37.9.9 RLIN3 Timing

Conditions: See **Section 37.1.1.1, Common Conditions**.

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---------------------|--------|---------------|------|------|------|------|
| RLIN3 transfer rate | | LIN function | 1 | | 20 | Kbps |
| | | UART function | | | 8 | Mbps |

37.9.10 FlexRay Timing

Conditions: See **Section 37.1.1.1, Common Conditions**.

Output signals are specified for the buffer "fast mode".

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---------------|--------|-----------|------|------|------|------|
| Transfer rate | | | | | 10 | Mbps |

37.9.11 PSI5 Timing

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|----------|--------|-----------|------|------|------|------|
| Bit time | (1) | 125 Kbps | 7.6 | 8.0 | 8.4 | μs |
| | (2) | 189 Kbps | 5.0 | 5.3 | 5.6 | μs |
| | (3) | 250 Kbps | | 4.0 | | μs |
| Gap time | (4) | 125 Kbps | 8.4 | | | μs |
| | (5) | 189 Kbps | 5.6 | | | μs |
| | (6) | 250 Kbps | 2.0 | | | μs |

37.9.12 RSENT Timing

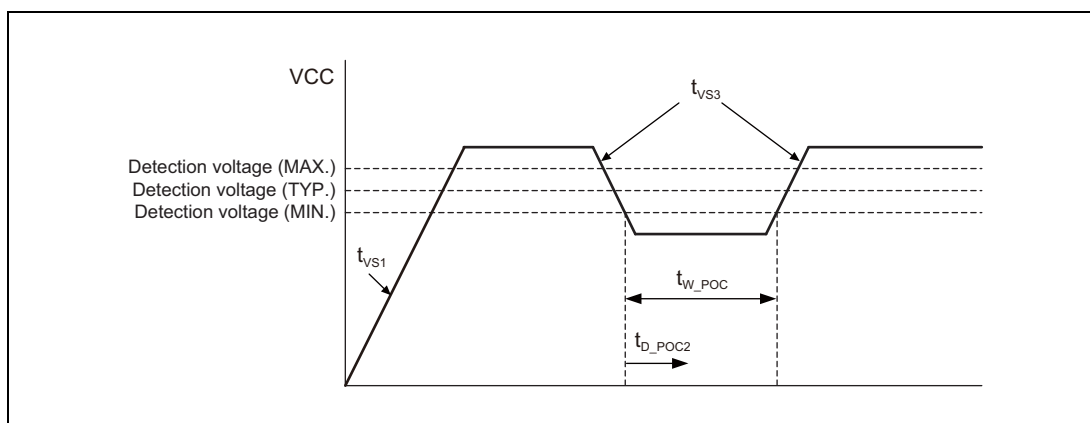
Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|-----------|--------|-----------|------|------|------|------|
| Tick Time | | | 1 | | 90 | μs |

37.10 POC Characteristics

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|-------------------|---------------|--|------|------|------|------|
| Detection voltage | VPOC | Rising | 2.7 | 2.85 | 3.0 | V |
| | | Falling | 2.7 | 2.8 | 2.9 | V |
| Response time 2 | t_{D_POC2} | After power on Voltage ramp (t_{VS3}) = 0.02 V/ms to 20 V/ms | | | 2 | ms |
| VCC minimum width | t_{W_POC} | | 0.2 | | | ms |
| VCC voltage ramp | t_{VS3} | | 0.02 | | 20 | V/ms |



37.11 Core Voltage Monitor Characteristics

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|----------------------------|--------------|-----------|------|------|------|---------|
| High VDD Detection voltage | VCVMH | | 1.35 | 1.40 | 1.45 | V |
| Low VDD Detection voltage | VCVML | | 1.10 | 1.15 | 1.20 | V |
| Response time | t_{D_CVM} | | | | 12 | μ s |

37.12 Temperature Sensor

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--------------------------------------|------------|-------------------------------------|------|------|------|------------------|
| Temperature precision | | $T_j = +25$ to $+150^\circ\text{C}$ | -6 | | 6 | $^\circ\text{C}$ |
| | | $T_j = -40$ to $+25^\circ\text{C}$ | -12 | | 12 | $^\circ\text{C}$ |
| Operation stabilization waiting time | t_{TSSB} | | 200 | | | μ s |

37.13 BIST Execution Time

Conditions: See Section 37.1.1.1, Common Conditions.

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|---------------------|--------|-----------|------|------|------|------|
| BIST execution time | | | | | 30 | ms |

37.14 A/D Converter Characteristics

Conditions: See Section 37.1.1.1, Common Conditions.

(1/2)

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit | | |
|---|---|---|---|---|---|---------|-----------|-----|
| Macro operation clock | | | | | 40 | MHz | | |
| Resolution | RESn | | | 12 | | bit | | |
| Conversion time | t_{CONn} | ADCGnSMPCR = 0000 _H , CLK_ADC = 40 MHz | | 1 | | μ s | | |
| | | ADCGnSMPCR = 90CC _H , CLK_ADC = 20 MHz | | 11.3 | | μ s | | |
| Total errors* ¹ | TOEn | AnVCC, AnVREFH = 4.3 V to 5.5 V, AnVCC – AnVREFH < 1V | ADCGnIm, When channel T&H is not used | | \pm 4.0 | LSB | | |
| | | | ADCGnIm, When channel T&H is used | | \pm 6.0 | LSB | | |
| | | AnVCC, AnVREFH = 3.0 V to 4.3 V, AnVCC – AnVREFH < 1V | ADCGnIm, When channel T&H is not used | | \pm 6.0 | LSB | | |
| | | | ADCGnIm, When channel T&H is used | | \pm 8.0 | LSB | | |
| | | Integral nonlinear * ¹ | ILEn | AnVCC, AnVREFH = 4.3 V to 5.5 V, AnVCC – AnVREFH < 1V | ADCGnIm, When channel T&H is not used | | \pm 2.0 | LSB |
| | | | | | ADCGnIm, When channel T&H is used | | \pm 3.0 | LSB |
| AnVCC, AnVREFH = 3.0 V to 4.3 V, AnVCC – AnVREFH < 1V | ADCGnIm, When channel T&H is not used | | | | \pm 3.0 | LSB | | |
| | ADCGnIm, When channel T&H is used | | \pm 4.0 | LSB | | | | |
| Differential nonlinear* ¹ | DLEn | AnVCC, AnVREFH = 4.3 V to 5.5 V, AnVCC – AnVREFH < 1V | ADCGnIm, When channel T&H is not used | | \pm 1.0 | LSB | | |
| | | | ADCGnIm, When channel T&H is used | | \pm 1.0 | LSB | | |
| | | AnVCC, AnVREFH = 3.0 V to 4.3 V, AnVCC – AnVREFH < 1V | ADCGnIm, When channel T&H is not used | | +2.0, –1.0 | LSB | | |
| ADCGnIm, When channel T&H is used | | | +3.0, –1.0 | LSB | | | | |

(2/2)

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--|--------------------|---|---|---------------|----------------------------------|------|
| Offset error*1 | ZSEn | AnVCC, AnVREFH = 4.3 V to 5.5 V, AnVCC – AnVREFH < 1V | ADCGnlm, When channel T&H is not used | | ±3.5 | LSB |
| | | | ADCGnlm, When channel T&H is used | | ±5.5 | LSB |
| | | AnVCC, AnVREFH = 3.0 V to 4.3 V, AnVCC – AnVREFH < 1V | ADCGnlm, When channel T&H is not used | | ±5.5 | LSB |
| | | | ADCGnlm, When channel T&H is used | | ±7.5 | LSB |
| Full-scale error*1 | FSEn | AnVCC, AnVREFH = 4.3 V to 5.5 V, AnVCC – AnVREFH < 1V | ADCGnlm, When channel T&H is not used | | ±3.5 | LSB |
| | | | ADCGnlm, When channel T&H is used | | ±5.5 | LSB |
| | | AnVCC, AnVREFH = 3.0 V to 4.3 V, AnVCC – AnVREFH < 1V | ADCGnlm, When channel T&H is not used | | ±5.5 | LSB |
| | | | ADCGnlm, When channel T&H is used | | ±7.5 | LSB |
| Analog input voltage | VAINmSN | AnVCC – AnVREFH < 1 V | When channel T&H is not used | AnVSS | AnVREFH | V |
| | | | When channel T&H is used | AnVSS + 0.2 V | AnVREFH –0.2 V | V |
| Sampling time | t _{SMP} | ADCGnSMPCR = 0000 _H , CLK_ADC = 40 MHz | | | t _{CONn} × (18/40) | μs |
| | | ADCGnSMPCR = 90CC _H , CLK_ADC = 20 MHz | | | t _{CONn} × (204/226) | μs |
| Channel T&H hold time*2 | t _{THOLD} | | | | 10 | μs |
| Pull-up resistor for A/D wiring break detection | RU_AIN | VIN = AnVSS | 10 | 20 | 40 | kΩ |
| Pull-down resistor for A/D wiring break | RD_AIN | VIN = AnVCC | 10 | 20 | 40 | kΩ |
| Total errors at self- diagnosis | TESH0SN | When A/D conversion circuit self-diagnostic function is used | | | ±40 | LSB |
| | | When pin-level self-diagnosis is used | AnVREFH = 4.3 V to 5.5 V | | ±80 | LSB |
| | | | AnVREFH = 3.0 V to 4.3 V | | ±170 | LSB |
| Equivalent input capacitance | CIN | During standby | | | 10 | pF |
| | | During sampling | | | 20 | pF |
| Allowable analog signal impedance | | Apply 0.1 μF to analog pin | | | 20 | kΩ |

Note 1. The values in the above table do not include quantization errors.

Note 2. Available retain time of sampling value by T&H circuit. Between sampling completion by T&H circuit and sampling completion of 12 bit SAR-AD by S/H circuit need to be executed within this time.

Note: n = 0,1 (number of units)
m = 0 to 11 (number of channels)

CAUTION

If current is injected to an analog input pin during the pin-level self-diagnosis, the conversion accuracy of the diagnostic voltage for the corresponding channel is not guaranteed.

37.15 Code Flash Characteristics

Table 37.11 Code Flash Basic Characteristics

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--|---------------|-------------------------------------|------|------|------|-------|
| Flash Sequencer Operating frequency | f_{CLKP_L} | | | | 40 | MHz |
| Programming endurance* ¹ | CWRT | Retained for 20 years* ² | 1000 | | | Times |
| Temperature range of programming | TPRG | T _j | -40 | | 150 | °C |
| Temperature range of reading | TREAD | T _j | -40 | | 150 | °C |

Note 1. Programming endurance is defined as the number of times each block is erased. Where programming endurance is n times (n = 1000 in this case), each block is erasable n times. For example, given a memory device that has 32-Kbyte erasure blocks, programming in the address range of each 256-byte programming block (128 programming operations, one for each block) in an erasure block and then erasing the block counts as one time in terms of programming endurance. However, programming of a given address range more than once after erasure is not possible (overwriting after programming is prohibited).

Note 2. This is the case when the average T_a is 85°C.

Table 37.12 Code Flash Programming Characteristics

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Condition | Block Size | MIN. | TYP. | MAX. | Unit |
|--------------------------------|--------------------------------------|------------|------|-------------------|-------------------|------|
| Programming time* ¹ | Programming endurance < 100 times | 256 B | | 2* ¹ | 6* ¹ | ms |
| | | 32 KB | | 200 | 360 | ms |
| | Programming endurance ≥ 100 times | 256 B | | 2.4* ¹ | 7.2* ¹ | ms |
| | | 32 KB | | 240 | 432 | ms |
| Erasing time* ¹ | Programming endurance < 100 times | 8 KB | | 50 | 120 | ms |
| | | 32 KB | | 200 | 480 | ms |
| | Programming endurance ≥ 100 times | 8 KB | | 60 | 144 | ms |
| | | 32 KB | | 240 | 576 | ms |

Note 1. Values are only for processing by hardware. Software overhead is not taken into account.

37.16 Data Flash Characteristics

Table 37.13 Data Flash Basic Characteristics

| Item | Symbol | Condition | MIN. | TYP. | MAX. | Unit |
|--|---------------|-------------------------------------|--------|------|------|-------|
| Flash Sequencer Operating frequency | f_{CLKP_L} | | | | 40 | MHz |
| Programming endurance ^{*1} | DVRT | Retained for 20 years ^{*2} | 125000 | | | Times |
| | | Retained for 3 years ^{*2} | 250000 | | | Times |
| Temperature range of programming | TPRG | Tj | -40 | | 150 | °C |
| Temperature range of reading | TREAD | Tj | -40 | | 150 | °C |

Note 1. Programming endurance is defined as the number of times each block is erased. Where programming endurance is n times (n = 125000 in this case), each block is erasable n times. For example, given a memory device that has 64-byte erasure blocks, programming in the address range of each 4-byte programming block (16 programming operations, one for each block) in an erasure block and then erasing the block counts as one time in terms of programming endurance. However, programming of a given address range more than once after erasure is not possible (overwriting after programming is prohibited).

Note 2. This is the case when the average Ta is 85°C.

Table 37.14 Data Flash Programing Characteristics

Conditions: See **Section 37.1.1.1, Common Conditions.**

| Item | Block Size | MIN. | TYP. | MAX. | Unit |
|------------------|------------|------|-------------------|-------------------|------|
| Programming time | 4 B | | 0.3 ^{*1} | 1.7 ^{*1} | ms |
| | 32 KB | | 2.5 | 6.8 | s |
| Erasing time | 64 B | | 3 ^{*1} | 10 ^{*1} | ms |
| | 32 KB | | 1.6 | 5.2 | s |
| Blank check time | 4 B | | | 30 ^{*1} | μs |
| | 64 B | | | 100 ^{*1} | μs |

Note 1. Values are only for processing by hardware. Software overhead is not taken into account.

37.17 Debug Interface

37.17.1 JTAG/Nexus Timing

Table 37.15 JTAG/Nexus Timing

Conditions: $T_j = -40^{\circ}\text{C}$ to 150°C , $CL = 30\text{ pF}$

| Item | Symbol | Condition | MIN. | MAX. | Unit |
|---|---------------------|-----------|------|------------------------|------|
| TCK cycle time | t_{TCKW} | | 40 | — | ns |
| TCK high level width | t_{TCKWH} | | 16 | — | ns |
| TCK low level width | t_{TCKWL} | | 16 | — | ns |
| TMS/TDI setup time (vs TCK↑) | t_{TISU} | | 12 | — | ns |
| TMS/TDI retaining time (vs TCK ↓) | t_{TIH} | | 12 | — | ns |
| TDO output delay time (vs TCK↓) | t_{TDOD} | | — | $t_{\text{TCKW}} - 20$ | ns |
| RDY output delay time (vs TCK ↓) | t_{RDYD} | | — | $t_{\text{TCKW}} - 20$ | ns |
| $\overline{\text{TRST}}$ low level width | t_{TRSTWL} | | 1500 | — | ns |
| TCK/ $\overline{\text{TRST}}$ /TMS/TDI input rising time | t_{TIR} | | — | 12 | ns |
| TCK/ $\overline{\text{TRST}}$ /TMS/TDI input falling time | t_{TIF} | | — | 12 | ns |

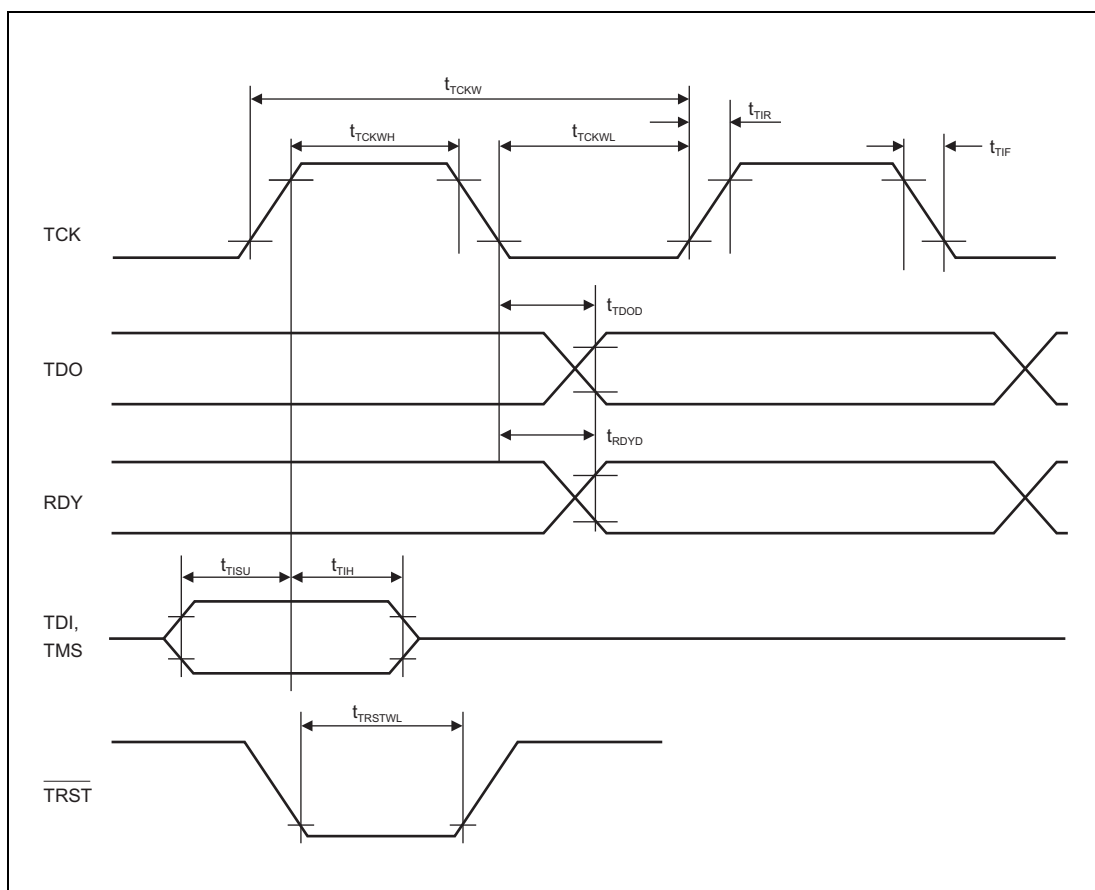


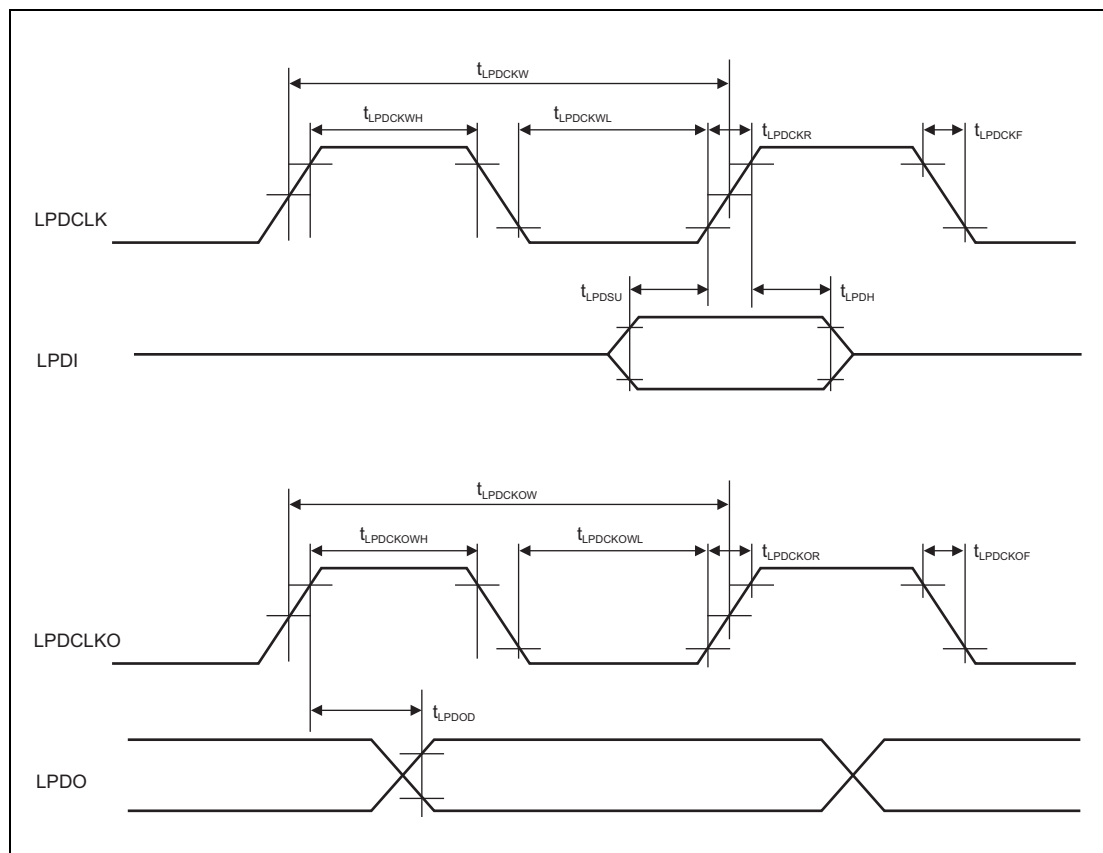
Figure 37.6 JTAG/Nexus Timing

37.17.2 LDU 4-Wire Timing

Table 37.16 LDU 4-Wire Timing

Conditions: $T_j = -40^{\circ}\text{C}$ to 150°C , $CL = 30\text{ pF}$

| Item | Symbol | Condition | MIN. | MAX. | Unit |
|---|----------------|-----------|---------------------|------|------|
| LPDCLK cycle time | t_{LPDCKW} | | 83.3 (max.12MHz) | — | ns |
| LPDCLK high level width | $t_{LPDCKWH}$ | | $t_{LPDCKW}/2 - 10$ | — | ns |
| LPDCLK low level width | $t_{LPDCKWL}$ | | $t_{LPDCKW}/2 - 10$ | — | ns |
| LPDCLK input rising time | t_{LPDCKR} | | — | 12 | ns |
| LPDCLK input falling time | t_{LPDCKF} | | — | 12 | ns |
| LPDI setup time (vs LPDCLK \uparrow) | t_{LPDSU} | | 41 | — | ns |
| LPDI retaining time (vs LPDCLK \uparrow) | t_{LPDH} | | 3 | — | ns |
| LPDCKO cycle time | $t_{LPDCKOW}$ | | 83.3 (max.12MHz) | — | ns |
| LPDCKO high level width | $t_{LPDCKOWH}$ | | $t_{LPDCKWH} - 12$ | — | ns |
| LPDCKO low level width | $t_{LPDCKOWL}$ | | $t_{LPDCKWL} - 12$ | — | ns |
| LPDCKO rising time | $t_{LPDCKOR}$ | | — | 12 | ns |
| LPDCKO falling time | $t_{LPDCKOF}$ | | — | 12 | ns |
| LPDO output delay (vs LPDCKO \uparrow) | t_{LPDOD} | | 0 | 15 | ns |



37.18 Thermal Characteristics

Table 37.17 Thermal Resistance of RH850/P1M-E

| Device | Parameter | Package | Condition | Estimated Value | Unit |
|--------|--------------|--------------------|---|-----------------|--------------------|
| P1M-E | ψ_{jb1} | LFQFP100 (14 × 14) | JEDEC, JESD51-7, $f_{CPLL} = 160\text{MHz}$, $T_J, \text{max} = 150^\circ\text{C}$ | 27.3 | $^\circ\text{C/W}$ |
| | ψ_{jb2} | LFQFP144 (16 × 16) | JEDEC, JESD51-7, $f_{CPLL} = 160\text{MHz}$, $T_J, \text{max} = 150^\circ\text{C}$ | 25.4 | $^\circ\text{C/W}$ |
| | ψ_{jb3} | LFQFP144 (20 × 20) | JEDEC, JESD51-7, $f_{CPLL} = 160\text{MHz}$, $T_J, \text{max} = 150^\circ\text{C}$ | 26.6 | $^\circ\text{C/W}$ |

Note: The parameter values of thermal resistance and characteristics vary depending on the usage environment.

Appendix A. List of Registers

Table A.1 List of Registers (1/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay Operation Control Register | FLXA0FROC | 00000000 _H | 32 | 10020004 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Operation Status Register | FLXA0FROS | 00000000 _H | 32 | 1002000C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Lock Register | FLXA0FRLCK | 00000000 _H | 32 | 1002001C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Error Interrupt Register | FLXA0FREIR | 00000000 _H | 32 | 10020020 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Status Interrupt Register | FLXA0FRSIR | 00000000 _H | 32 | 10020024 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Error Interrupt Line Select | FLXA0FREILS | 00000000 _H | 32 | 10020028 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Status Interrupt Line Select | FLXA0FRSILS | 0303FFFF _H | 32 | 1002002C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Error Interrupt Enable Set Register | FLXA0FREIES | 00000000 _H | 32 | 10020030 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Error Interrupt Enable Reset Register | FLXA0FREIER | 00000000 _H | 32 | 10020034 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Status Interrupt Enable Set Register | FLXA0FRSIES | 00000000 _H | 32 | 10020038 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Status Interrupt Enable Reset Register | FLXA0FRSIER | 00000000 _H | 32 | 1002003C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Interrupt Line Enable Register | FLXA0FRILE | 00000000 _H | 32 | 10020040 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Timer 0 Configuration Register | FLXA0FRTOC | 00000000 _H | 32 | 10020044 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Timer 1 Configuration Register | FLXA0FRT1C | 00020000 _H | 32 | 10020048 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Stop Watch Register 1 | FLXA0FRSTPW1 | 00000000 _H | 32 | 1002004C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Stop Watch Register 2 | FLXA0FRSTPW2 | 00000000 _H | 32 | 10020050 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay SUC Configuration Register 1 | FLXA0FRSUCC1 | 0C401080 _H | 32 | 10020080 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay SUC Configuration Register 2 | FLXA0FRSUCC2 | 01000504 _H | 32 | 10020084 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay SUC Configuration Register 3 | FLXA0FRSUCC3 | 00000011 _H | 32 | 10020088 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay NEM Configuration Register | FLXA0FRNEMC | 00000000 _H | 32 | 1002008C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay PRT Configuration Register 1 | FLXA0FRPRTC1 | 084C0633 _H | 32 | 10020090 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay PRT Configuration Register 2 | FLXA0FRPRTC2 | 0F2D0A0E _H | 32 | 10020094 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay MHD Configuration Register | FLXA0FRMHDC | 00000000 _H | 32 | 10020098 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 1 | FLXA0FRGTUC1 | 00000280 _H | 32 | 100200A0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 2 | FLXA0FRGTUC2 | 0002000A _H | 32 | 100200A4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 3 | FLXA0FRGTUC3 | 02020000 _H | 32 | 100200A8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 4 | FLXA0FRGTUC4 | 00080007 _H | 32 | 100200AC _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 5 | FLXA0FRGTUC5 | 0E000000 _H | 32 | 100200B0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 6 | FLXA0FRGTUC6 | 00020000 _H | 32 | 100200B4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 7 | FLXA0FRGTUC7 | 00020004 _H | 32 | 100200B8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 8 | FLXA0FRGTUC8 | 00000002 _H | 32 | 100200BC _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 9 | FLXA0FRGTUC9 | 00000101 _H | 32 | 100200C0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 10 | FLXA0FRGTUC10 | 00020005 _H | 32 | 100200C4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay GTU Configuration Register 11 | FLXA0FRGTUC11 | 00000000 _H | 32 | 100200C8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (2/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay CC Status Vector Register | FLXA0FRCCSV | 00104000 _H | 32 | 10020100 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay CC Error Vector Register | FLXA0FRCEV | 00000000 _H | 32 | 10020104 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Slot Counter Value Register | FLXA0FRSCV | 00000000 _H | 32 | 10020110 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Macrotick and Cycle Counter Value Register | FLXA0FRMTCCV | 00000000 _H | 32 | 10020114 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Rate Correction Value Register | FLXA0FRRCV | 00000000 _H | 32 | 10020118 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Offset Correction Value Register | FLXA0FROCV | 00000000 _H | 32 | 1002011C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Sync Frame Status Register | FLXA0FRSFS | 00000000 _H | 32 | 10020120 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Symbol Window and NIT Status Register | FLXA0FRSWNIT | 00000000 _H | 32 | 10020124 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Aggregated Channel Status Register | FLXA0FRACS | 00000000 _H | 32 | 10020128 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 1 | FLXA0FRESID1 | 00000000 _H | 32 | 10020130 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 2 | FLXA0FRESID2 | 00000000 _H | 32 | 10020134 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 3 | FLXA0FRESID3 | 00000000 _H | 32 | 10020138 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 4 | FLXA0FRESID4 | 00000000 _H | 32 | 1002013C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 5 | FLXA0FRESID5 | 00000000 _H | 32 | 10020140 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 6 | FLXA0FRESID6 | 00000000 _H | 32 | 10020144 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 7 | FLXA0FRESID7 | 00000000 _H | 32 | 10020148 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 8 | FLXA0FRESID8 | 00000000 _H | 32 | 1002014C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 9 | FLXA0FRESID9 | 00000000 _H | 32 | 10020150 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 10 | FLXA0FRESID10 | 00000000 _H | 32 | 10020154 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 11 | FLXA0FRESID11 | 00000000 _H | 32 | 10020158 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 12 | FLXA0FRESID12 | 00000000 _H | 32 | 1002015C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 13 | FLXA0FRESID13 | 00000000 _H | 32 | 10020160 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 14 | FLXA0FRESID14 | 00000000 _H | 32 | 10020164 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Even Sync ID Register 15 | FLXA0FRESID15 | 00000000 _H | 32 | 10020168 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 1 | FLXA0FROSID1 | 00000000 _H | 32 | 10020170 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 2 | FLXA0FROSID2 | 00000000 _H | 32 | 10020174 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 3 | FLXA0FROSID3 | 00000000 _H | 32 | 10020178 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 4 | FLXA0FROSID4 | 00000000 _H | 32 | 1002017C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 5 | FLXA0FROSID5 | 00000000 _H | 32 | 10020180 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 6 | FLXA0FROSID6 | 00000000 _H | 32 | 10020184 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 7 | FLXA0FROSID7 | 00000000 _H | 32 | 10020188 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 8 | FLXA0FROSID8 | 00000000 _H | 32 | 1002018C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 9 | FLXA0FROSID9 | 00000000 _H | 32 | 10020190 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 10 | FLXA0FROSID10 | 00000000 _H | 32 | 10020194 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 11 | FLXA0FROSID11 | 00000000 _H | 32 | 10020198 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Odd Sync ID Register 12 | FLXA0FROSID12 | 00000000 _H | 32 | 1002019C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (3/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay Odd Sync ID Register 13 | FLXA0FROSID13 | 00000000 _H | 32 | 100201A0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Odd Sync ID Register 14 | FLXA0FROSID14 | 00000000 _H | 32 | 100201A4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Odd Sync ID Register 15 | FLXA0FROSID15 | 00000000 _H | 32 | 100201A8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Network Management Vector Register 1 | FLXA0FRNMV1 | 00000000 _H | 32 | 100201B0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Network Management Vector Register 2 | FLXA0FRNMV2 | 00000000 _H | 32 | 100201B4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Network Management Vector Register 3 | FLXA0FRNMV3 | 00000000 _H | 32 | 100201B8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message RAM Configuration Register | FLXA0FRMRC | 01800000 _H | 32 | 10020300 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay FIFO Rejection Filter Register | FLXA0FRFRF | 01800000 _H | 32 | 10020304 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay FIFO Rejection Filter Mask Register | FLXA0FRFRFM | 00000000 _H | 32 | 10020308 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay FIFO Critical Level Register | FLXA0FRFCL | 00000080 _H | 32 | 1002030C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message Handler Status Register | FLXA0FRMHDS | 00000080 _H | 32 | 10020310 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Last Dynamic Transmit Slot Register | FLXA0FRLDTS | 00000000 _H | 32 | 10020314 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay FIFO Status Register | FLXA0FRFSR | 00000000 _H | 32 | 10020318 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message Handler Constraints Flags Register | FLXA0FRMHDF | 00000000 _H | 32 | 1002031C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Transmission Register 1 | FLXA0FRTXRQ1 | 00000000 _H | 32 | 10020320 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Transmission Register 2 | FLXA0FRTXRQ2 | 00000000 _H | 32 | 10020324 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Transmission Register 3 | FLXA0FRTXRQ3 | 00000000 _H | 32 | 10020328 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Transmission Register 4 | FLXA0FRTXRQ4 | 00000000 _H | 32 | 1002032C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay New Data Register 1 | FLXA0FRNDAT1 | 00000000 _H | 32 | 10020330 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay New Data Register 2 | FLXA0FRNDAT2 | 00000000 _H | 32 | 10020334 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay New Data Register 3 | FLXA0FRNDAT3 | 00000000 _H | 32 | 10020338 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay New Data Register 4 | FLXA0FRNDAT4 | 00000000 _H | 32 | 1002033C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message Buffer Status Changed Register 1 | FLXA0FRMBSC1 | 00000000 _H | 32 | 10020340 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message Buffer Status Changed Register 2 | FLXA0FRMBSC2 | 00000000 _H | 32 | 10020344 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message Buffer Status Changed Register 3 | FLXA0FRMBSC3 | 00000000 _H | 32 | 10020348 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message Buffer Status Changed Register 4 | FLXA0FRMBSC4 | 00000000 _H | 32 | 1002034C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 1 | FLXA0FRWRDS1 | 00000000 _H | 32 | 10020400 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 2 | FLXA0FRWRDS2 | 00000000 _H | 32 | 10020404 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 3 | FLXA0FRWRDS3 | 00000000 _H | 32 | 10020408 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 4 | FLXA0FRWRDS4 | 00000000 _H | 32 | 1002040C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 5 | FLXA0FRWRDS5 | 00000000 _H | 32 | 10020410 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 6 | FLXA0FRWRDS6 | 00000000 _H | 32 | 10020414 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 7 | FLXA0FRWRDS7 | 00000000 _H | 32 | 10020418 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 8 | FLXA0FRWRDS8 | 00000000 _H | 32 | 1002041C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 9 | FLXA0FRWRDS9 | 00000000 _H | 32 | 10020420 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 10 | FLXA0FRWRDS10 | 00000000 _H | 32 | 10020424 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (4/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay Write Data Section Register 11 | FLXA0FRWRDS11 | 00000000 _H | 32 | 10020428 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 12 | FLXA0FRWRDS12 | 00000000 _H | 32 | 1002042C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 13 | FLXA0FRWRDS13 | 00000000 _H | 32 | 10020430 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 14 | FLXA0FRWRDS14 | 00000000 _H | 32 | 10020434 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 15 | FLXA0FRWRDS15 | 00000000 _H | 32 | 10020438 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 16 | FLXA0FRWRDS16 | 00000000 _H | 32 | 1002043C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 17 | FLXA0FRWRDS17 | 00000000 _H | 32 | 10020440 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 18 | FLXA0FRWRDS18 | 00000000 _H | 32 | 10020444 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 19 | FLXA0FRWRDS19 | 00000000 _H | 32 | 10020448 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 20 | FLXA0FRWRDS20 | 00000000 _H | 32 | 1002044C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 21 | FLXA0FRWRDS21 | 00000000 _H | 32 | 10020450 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 22 | FLXA0FRWRDS22 | 00000000 _H | 32 | 10020454 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 23 | FLXA0FRWRDS23 | 00000000 _H | 32 | 10020458 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 24 | FLXA0FRWRDS24 | 00000000 _H | 32 | 1002045C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 25 | FLXA0FRWRDS25 | 00000000 _H | 32 | 10020460 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 26 | FLXA0FRWRDS26 | 00000000 _H | 32 | 10020464 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 27 | FLXA0FRWRDS27 | 00000000 _H | 32 | 10020468 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 28 | FLXA0FRWRDS28 | 00000000 _H | 32 | 1002046C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 29 | FLXA0FRWRDS29 | 00000000 _H | 32 | 10020470 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 30 | FLXA0FRWRDS30 | 00000000 _H | 32 | 10020474 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 31 | FLXA0FRWRDS31 | 00000000 _H | 32 | 10020478 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 32 | FLXA0FRWRDS32 | 00000000 _H | 32 | 1002047C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 33 | FLXA0FRWRDS33 | 00000000 _H | 32 | 10020480 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 34 | FLXA0FRWRDS34 | 00000000 _H | 32 | 10020484 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 35 | FLXA0FRWRDS35 | 00000000 _H | 32 | 10020488 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 36 | FLXA0FRWRDS36 | 00000000 _H | 32 | 1002048C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 37 | FLXA0FRWRDS37 | 00000000 _H | 32 | 10020490 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 38 | FLXA0FRWRDS38 | 00000000 _H | 32 | 10020494 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 39 | FLXA0FRWRDS39 | 00000000 _H | 32 | 10020498 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 40 | FLXA0FRWRDS40 | 00000000 _H | 32 | 1002049C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 41 | FLXA0FRWRDS41 | 00000000 _H | 32 | 100204A0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 42 | FLXA0FRWRDS42 | 00000000 _H | 32 | 100204A4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 43 | FLXA0FRWRDS43 | 00000000 _H | 32 | 100204A8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 44 | FLXA0FRWRDS44 | 00000000 _H | 32 | 100204AC _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 45 | FLXA0FRWRDS45 | 00000000 _H | 32 | 100204B0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Write Data Section Register 46 | FLXA0FRWRDS46 | 00000000 _H | 32 | 100204B4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (5/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay Write Data Section Register 47 | FLXA0FRWRDS47 | 00000000 _H | 32 | 100204B8 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 48 | FLXA0FRWRDS48 | 00000000 _H | 32 | 100204BC _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 49 | FLXA0FRWRDS49 | 00000000 _H | 32 | 100204C0 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 50 | FLXA0FRWRDS50 | 00000000 _H | 32 | 100204C4 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 51 | FLXA0FRWRDS51 | 00000000 _H | 32 | 100204C8 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 52 | FLXA0FRWRDS52 | 00000000 _H | 32 | 100204CC _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 53 | FLXA0FRWRDS53 | 00000000 _H | 32 | 100204D0 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 54 | FLXA0FRWRDS54 | 00000000 _H | 32 | 100204D4 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 55 | FLXA0FRWRDS55 | 00000000 _H | 32 | 100204D8 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 56 | FLXA0FRWRDS56 | 00000000 _H | 32 | 100204DC _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 57 | FLXA0FRWRDS57 | 00000000 _H | 32 | 100204E0 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 58 | FLXA0FRWRDS58 | 00000000 _H | 32 | 100204E4 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 59 | FLXA0FRWRDS59 | 00000000 _H | 32 | 100204E8 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 60 | FLXA0FRWRDS60 | 00000000 _H | 32 | 100204EC _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 61 | FLXA0FRWRDS61 | 00000000 _H | 32 | 100204F0 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 62 | FLXA0FRWRDS62 | 00000000 _H | 32 | 100204F4 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 63 | FLXA0FRWRDS63 | 00000000 _H | 32 | 100204F8 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Data Section Register 64 | FLXA0FRWRDS64 | 00000000 _H | 32 | 100204FC _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Header Section Register 1 | FLXA0FRWRHS1 | 00000000 _H | 32 | 10020500 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Header Section Register 2 | FLXA0FRWRHS2 | 00000000 _H | 32 | 10020504 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Write Header Section Register 3 | FLXA0FRWRHS3 | 00000000 _H | 32 | 10020508 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Input Buffer Command Mask Register | FLXA0FRIBCM | 00000000 _H | 32 | 10020510 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Input Buffer Command Request Register | FLXA0FRIBCR | 00000000 _H | 32 | 10020514 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 1 | FLXA0FRRDDS1 | 00000000 _H | 32 | 10020600 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 2 | FLXA0FRRDDS2 | 00000000 _H | 32 | 10020604 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 3 | FLXA0FRRDDS3 | 00000000 _H | 32 | 10020608 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 4 | FLXA0FRRDDS4 | 00000000 _H | 32 | 1002060C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 5 | FLXA0FRRDDS5 | 00000000 _H | 32 | 10020610 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 6 | FLXA0FRRDDS6 | 00000000 _H | 32 | 10020614 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 7 | FLXA0FRRDDS7 | 00000000 _H | 32 | 10020618 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 8 | FLXA0FRRDDS8 | 00000000 _H | 32 | 1002061C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 9 | FLXA0FRRDDS9 | 00000000 _H | 32 | 10020620 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 10 | FLXA0FRRDDS10 | 00000000 _H | 32 | 10020624 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 11 | FLXA0FRRDDS11 | 00000000 _H | 32 | 10020628 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 12 | FLXA0FRRDDS12 | 00000000 _H | 32 | 1002062C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 13 | FLXA0FRRDDS13 | 00000000 _H | 32 | 10020630 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (6/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---------------------------------------|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay Read Data Section Register 14 | FLXA0FRRDDS14 | 00000000 _H | 32 | 10020634 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 15 | FLXA0FRRDDS15 | 00000000 _H | 32 | 10020638 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 16 | FLXA0FRRDDS16 | 00000000 _H | 32 | 1002063C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 17 | FLXA0FRRDDS17 | 00000000 _H | 32 | 10020640 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 18 | FLXA0FRRDDS18 | 00000000 _H | 32 | 10020644 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 19 | FLXA0FRRDDS19 | 00000000 _H | 32 | 10020648 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 20 | FLXA0FRRDDS20 | 00000000 _H | 32 | 1002064C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 21 | FLXA0FRRDDS21 | 00000000 _H | 32 | 10020650 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 22 | FLXA0FRRDDS22 | 00000000 _H | 32 | 10020654 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 23 | FLXA0FRRDDS23 | 00000000 _H | 32 | 10020658 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 24 | FLXA0FRRDDS24 | 00000000 _H | 32 | 1002065C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 25 | FLXA0FRRDDS25 | 00000000 _H | 32 | 10020660 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 26 | FLXA0FRRDDS26 | 00000000 _H | 32 | 10020664 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 27 | FLXA0FRRDDS27 | 00000000 _H | 32 | 10020668 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 28 | FLXA0FRRDDS28 | 00000000 _H | 32 | 1002066C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 29 | FLXA0FRRDDS29 | 00000000 _H | 32 | 10020670 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 30 | FLXA0FRRDDS30 | 00000000 _H | 32 | 10020674 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 31 | FLXA0FRRDDS31 | 00000000 _H | 32 | 10020678 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 32 | FLXA0FRRDDS32 | 00000000 _H | 32 | 1002067C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 33 | FLXA0FRRDDS33 | 00000000 _H | 32 | 10020680 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 34 | FLXA0FRRDDS34 | 00000000 _H | 32 | 10020684 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 35 | FLXA0FRRDDS35 | 00000000 _H | 32 | 10020688 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 36 | FLXA0FRRDDS36 | 00000000 _H | 32 | 1002068C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 37 | FLXA0FRRDDS37 | 00000000 _H | 32 | 10020690 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 38 | FLXA0FRRDDS38 | 00000000 _H | 32 | 10020694 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 39 | FLXA0FRRDDS39 | 00000000 _H | 32 | 10020698 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 40 | FLXA0FRRDDS40 | 00000000 _H | 32 | 1002069C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 41 | FLXA0FRRDDS41 | 00000000 _H | 32 | 100206A0 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 42 | FLXA0FRRDDS42 | 00000000 _H | 32 | 100206A4 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 43 | FLXA0FRRDDS43 | 00000000 _H | 32 | 100206A8 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 44 | FLXA0FRRDDS44 | 00000000 _H | 32 | 100206AC _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 45 | FLXA0FRRDDS45 | 00000000 _H | 32 | 100206B0 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 46 | FLXA0FRRDDS46 | 00000000 _H | 32 | 100206B4 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 47 | FLXA0FRRDDS47 | 00000000 _H | 32 | 100206B8 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 48 | FLXA0FRRDDS48 | 00000000 _H | 32 | 100206BC _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Read Data Section Register 49 | FLXA0FRRDDS49 | 00000000 _H | 32 | 100206C0 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (7/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay Read Data Section Register 50 | FLXA0FRDDS50 | 00000000 _H | 32 | 100206C4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 51 | FLXA0FRDDS51 | 00000000 _H | 32 | 100206C8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 52 | FLXA0FRDDS52 | 00000000 _H | 32 | 100206CC _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 53 | FLXA0FRDDS53 | 00000000 _H | 32 | 100206D0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 54 | FLXA0FRDDS54 | 00000000 _H | 32 | 100206D4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 55 | FLXA0FRDDS55 | 00000000 _H | 32 | 100206D8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 56 | FLXA0FRDDS56 | 00000000 _H | 32 | 100206DC _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 57 | FLXA0FRDDS57 | 00000000 _H | 32 | 100206E0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 58 | FLXA0FRDDS58 | 00000000 _H | 32 | 100206E4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 59 | FLXA0FRDDS59 | 00000000 _H | 32 | 100206E8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 60 | FLXA0FRDDS60 | 00000000 _H | 32 | 100206EC _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 61 | FLXA0FRDDS61 | 00000000 _H | 32 | 100206F0 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 62 | FLXA0FRDDS62 | 00000000 _H | 32 | 100206F4 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 63 | FLXA0FRDDS63 | 00000000 _H | 32 | 100206F8 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Data Section Register 64 | FLXA0FRDDS64 | 00000000 _H | 32 | 100206FC _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Header Section Register 1 | FLXA0FRDHS1 | 00000000 _H | 32 | 10020700 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Header Section Register 2 | FLXA0FRDHS2 | 00000000 _H | 32 | 10020704 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Read Header Section Register 3 | FLXA0FRDHS3 | 00000000 _H | 32 | 10020708 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Message Buffer Status Register | FLXA0FRMBS | 00000000 _H | 32 | 1002070C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Output Buffer Command Mask Register | FLXA0FROBCM | 00000000 _H | 32 | 10020710 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Output Buffer Command Request Register | FLXA0FROBCR | 00000000 _H | 32 | 10020714 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Input Transfer Configuration Register | FLXA0FRITC | 00000000 _H | 32 | 10020800 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Output Transfer Configuration Register | FLXA0FROTC | 00000000 _H | 32 | 10020804 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Input pointer table Base Address Register | FLXA0FRIBA | 00000000 _H | 32 | 10020808 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay FIFO pointer table Base Address Register | FLXA0FRFBA | 00000000 _H | 32 | 1002080C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Output pointer table Base Address Register | FLXA0FROBA | 00000000 _H | 32 | 10020810 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Input Queue Control Register | FLXA0FRIQC | 00000000 _H | 32 | 10020814 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay User Input transfer Request Register | FLXA0FRUIR | 00000000 _H | 32 | 10020818 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay User Output transfer Request Register | FLXA0FRUOR | 00000000 _H | 32 | 1002081C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Input Transfer Status Register | FLXA0FRITS | 00000000 _H | 32 | 10020820 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Output Transfer Status Register | FLXA0FROTS | 00000000 _H | 32 | 10020824 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Access Error Status Register | FLXA0FRAES | 00000000 _H | 32 | 10020828 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay Access Error Address Register | FLXA0FRAEA | 00000000 _H | 32 | 1002082C _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay message Data Available Register 0 | FLXA0FRDA0 | 00000000 _H | 32 | 10020830 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay message Data Available Register 1 | FLXA0FRDA1 | 00000000 _H | 32 | 10020834 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| FLXA0 | FlexRay message Data Available Register 2 | FLXA0FRDA2 | 00000000 _H | 32 | 10020838 _H | H-Bus (FlexRay) | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (8/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FLXA0 | FlexRay message Data Available Register 3 | FLXA0FRDA3 | 00000000 _H | 32 | 1002083C _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay H-Bus Configuration Register | FLXA0FRAHBC | 00000000 _H | 32 | 10020840 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLXA0 | FlexRay Timer 2 Configuration Register | FLXA0FRT2C | 00000000 _H | 32 | 10020844 _H | H-Bus (FlexRay) | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLMD | FLMDCNT Register | FLMDCNT | 00000000 _H | 32 | FFA00000 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FLMD | FLMD Write-Protection Command Register | FLMDPCMD | 00000000 _H | 32 | FFA00004 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FLMD | FLMD Write Sequence Status Register | FLMDPS | 00000000 _H | 32 | FFA00008 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FLASH | Self-programming ID input register 0 | SELFID0 | 00000000 _H | 32 | FFA08000 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FLASH | Self-programming ID input register 0 | SELFID1 | 00000000 _H | 32 | FFA08004 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FLASH | Self-programming ID input register 0 | SELFID2 | 00000000 _H | 32 | FFA08008 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FLASH | Self-programming ID input register 0 | SELFID3 | 00000000 _H | 32 | FFA0800C _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FLASH | Self-programming ID authentication status register | SELFIDST | 0000000X _H | 32 | FFA08010 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash pin monitoring register | FPMON | X0 _H | 8 | FFA10000 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash access status register | FASTAT | 00 _H | 8 | FFA10010 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash access error interrupt enable register | FAEINT | 99 _H | 8 | FFA10014 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Code Flash memory area selecting register | FAEASELC | 0000 _H | 16 | FFA10020 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | FACI command start address register | FSADDR | 00000000 _H | 32 | FFA10030 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | FACI command end address register | FEADDR | 00000000 _H | 32 | FFA10034 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash status register | FSTATR | 00008000 _H | 32 | FFA10080 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash P/E protection register | FENTRYR | 0000 _H | 16 | FFA10084 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Code Flash protection register | FPROTR | 0000 _H | 16 | FFA10088 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash Sequencer Set-Up Initialize Register | FSUINTR | 0000 _H | 16 | FFA1008C _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Lock bit status register | FLKSTAT | 00 _H | 8 | FFA10090 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | FACI reset transfer status register | FRTSTAT | 0X _H | 8 | FFA10098 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | FACI reset transfer error interrupt enable register | FRTINT | 03 _H | 8 | FFA1009C _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | FACI command register | FCMDR | FFFF _H | 16 | FFA100A0 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash P/E status register | FPESTAT | 0000 _H | 16 | FFA100C0 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Data Flash blank check control register | FBCCNT | 00 _H | 8 | FFA100D0 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Data Flash blank check status register | FBCSTAT | 00 _H | 8 | FFA100D4 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Data Flash programming start address register | FPSADDR | 00000000 _H | 32 | FFA100D8 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash sequencer process switch register | FCPSR | 0000 _H | 16 | FFA100E0 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash sequencer processing clock notify register | FPCKAR | 0028 _H | 16 | FFA100E4 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash emulation control register | FLEMU | 00 _H | 8 | FFA100F0 _H | 5 | 8 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash emulation address specify register | FLEAD | 00000000 _H | 32 | FFA100F4 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash ECC encoder monitor register | FECCEMON | FFFF _H | 16 | FFA10100 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash ECC test mode register | FECCTMD | 0030 _H | 16 | FFA10104 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| FACI | Flash dummy ECC register | FDMYECC | FFFF _H | 16 | FFA10108 _H | 5 | 16 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| SINT | Software Interrupt Registers | SINTR0 | 00 _H | 8 | FFC00000 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SINT | Software Interrupt Registers | SINTR1 | 00 _H | 8 | FFC00004 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SINT | Software Interrupt Registers | SINTR2 | 00 _H | 8 | FFC00008 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SINT | Software Interrupt Registers | SINTR3 | 00 _H | 8 | FFC0000C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SINT | Software Interrupt Registers | SINTR4 | 00 _H | 8 | FFC00010 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| BRAM | Backup register 0 | BRAMDAT0 | Undefined (retained) | 32 | FFC0A000 _H | 5 | 32 | — | — | — | — | ✓ | ✓ |
| BRAM | Backup register 1 | BRAMDAT1 | Undefined (retained) | 32 | FFC0A004 _H | 5 | 32 | — | — | — | — | ✓ | ✓ |
| BRAM | Backup register 2 | BRAMDAT2 | Undefined (retained) | 32 | FFC0A008 _H | 5 | 32 | — | — | — | — | ✓ | ✓ |
| BRAM | Backup register 3 | BRAMDAT3 | Undefined (retained) | 32 | FFC0A00C _H | 5 | 32 | — | — | — | — | ✓ | ✓ |

Table A.1 List of Registers (9/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PORT | Port register | P0 | 0000 _H | 16 | FFC10000 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port set/reset register | PSR0 | 00000000 _H | 32 | FFC10004 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port NOT register | PNOT0 | 0000 _H | 16 | FFC10008 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port pin read register | PPR0 | 0000 _H | 16 | FFC1000C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode register | PM0 | FBFF _H | 16 | FFC10010 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control register | PMC0 | 0000 _H | 16 | FFC10014 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control register | PFC0 | 0000 _H | 16 | FFC10018 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control expansion register | PFCE0 | 0000 _H | 16 | FFC1001C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode set/reset register | PMSR0 | 0000FBFF _H | 32 | FFC10020 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control set/reset register | PMCSR0 | 00000000 _H | 32 | FFC10024 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control addition expansion register | PFCAE0 | 0000 _H | 16 | FFC10028 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port output level inversion register | PINV0 | 00000000 _H | 32 | FFC10030 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port register | P1 | 0000 _H | 16 | FFC10040 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port set/reset register | PSR1 | 00000000 _H | 32 | FFC10044 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port NOT register | PNOT1 | 0000 _H | 16 | FFC10048 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port pin read register | PPR1 | 0000 _H | 16 | FFC1004C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode register | PM1 | FFFF _H | 16 | FFC10050 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control register | PMC1 | 0000 _H | 16 | FFC10054 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control register | PFC1 | 0000 _H | 16 | FFC10058 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control expansion register | PFCE1 | 0000 _H | 16 | FFC1005C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode set/reset register | PMSR1 | 0000FFFF _H | 32 | FFC10060 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control set/reset register | PMCSR1 | 00000000 _H | 32 | FFC10064 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control addition expansion register | PFCAE1 | 0000 _H | 16 | FFC10068 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port output level inversion register | PINV1 | 00000000 _H | 32 | FFC10070 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port register | P2 | 0000 _H | 16 | FFC10080 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port set/reset register | PSR2 | 00000000 _H | 32 | FFC10084 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port NOT register | PNOT2 | 0000 _H | 16 | FFC10088 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port pin read register | PPR2 | 0000 _H | 16 | FFC1008C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode register | PM2 | FFFF _H | 16 | FFC10090 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control register | PMC2 | 0000 _H | 16 | FFC10094 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control register | PFC2 | 0000 _H | 16 | FFC10098 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control expansion register | PFCE2 | 0000 _H | 16 | FFC1009C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode set/reset register | PMSR2 | 0000FFFF _H | 32 | FFC100A0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control set/reset register | PMCSR2 | 00000000 _H | 32 | FFC100A4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control addition expansion register | PFCAE2 | 0000 _H | 16 | FFC100A8 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port output level inversion register | PINV2 | 00000000 _H | 32 | FFC100B0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port register | P3 | 0000 _H | 16 | FFC100C0 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port set/reset register | PSR3 | 00000000 _H | 32 | FFC100C4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port NOT register | PNOT3 | 0000 _H | 16 | FFC100C8 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port pin read register | PPR3 | 0000 _H | 16 | FFC100CC _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode register | PM3 | FFFF _H | 16 | FFC100D0 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control register | PMC3 | 0000 _H | 16 | FFC100D4 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control register | PFC3 | 0000 _H | 16 | FFC100D8 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control expansion register | PFCE3 | 0000 _H | 16 | FFC100DC _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode set/reset register | PMSR3 | 0000FFFF _H | 32 | FFC100E0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port mode control set/reset register | PMCSR3 | 00000000 _H | 32 | FFC100E4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port function control addition expansion register | PFCAE3 | 0000 _H | 16 | FFC100E8 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port output level inversion register | PINV3 | 00000000 _H | 32 | FFC100F0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (10/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PORT | Port register | P4 | 0000 _H | 16 | FFC10100 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port set/reset register | PSR4 | 00000000 _H | 32 | FFC10104 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port NOT register | PNOT4 | 0000 _H | 16 | FFC10108 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port pin read register | PPR4 | 0000 _H | 16 | FFC1010C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode register | PM4 | FFFF _H | 16 | FFC10110 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode control register | PMC4 | 0000 _H | 16 | FFC10114 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port function control register | PFC4 | 0000 _H | 16 | FFC10118 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port function control expansion register | PFCE4 | 0000 _H | 16 | FFC1011C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode set/reset register | PMSR4 | 0000FFFF _H | 32 | FFC10120 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode control set/reset register | PMCSR4 | 00000000 _H | 32 | FFC10124 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port function control addition expansion register | PFCAE4 | 0000 _H | 16 | FFC10128 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port output level inversion register | PINV4 | 00000000 _H | 32 | FFC10130 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port register | P5 | 0000 _H | 16 | FFC10140 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port set/reset register | PSR5 | 00000000 _H | 32 | FFC10144 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port NOT register | PNOT5 | 0000 _H | 16 | FFC10148 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port pin read register | PPR5 | 0000 _H | 16 | FFC1014C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode register | PM5 | FFFF _H | 16 | FFC10150 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode control register | PMC5 | 0000 _H | 16 | FFC10154 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port function control register | PFC5 | 0000 _H | 16 | FFC10158 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port function control expansion register | PFCE5 | 0000 _H | 16 | FFC1015C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode set/reset register | PMSR5 | 0000FFFF _H | 32 | FFC10160 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port mode control set/reset register | PMCSR5 | 00000000 _H | 32 | FFC10164 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port function control addition expansion register | PFCAE5 | 0000 _H | 16 | FFC10168 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port output level inversion register | PINV5 | 00000000 _H | 32 | FFC10170 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_0 | 00000010 _H | 32 | FFC12000 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_1 | 00000010 _H | 32 | FFC12004 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_2 | 00000010 _H | 32 | FFC12008 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_3 | 00000010 _H | 32 | FFC1200C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_4 | 00000010 _H | 32 | FFC12010 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_5 | 00000010 _H | 32 | FFC12014 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_6 | 00000010 _H | 32 | FFC12018 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_7 | 00000010 _H | 32 | FFC1201C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_8 | 00000010 _H | 32 | FFC12020 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_9 | 00000010 _H | 32 | FFC12024 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_10 | 11400000 _H | 32 | FFC12028 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_11 | 00000010 _H | 32 | FFC1202C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_12 | 00000010 _H | 32 | FFC12030 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_13 | 00000010 _H | 32 | FFC12034 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR0_14 | 00000010 _H | 32 | FFC12038 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR1_0 | 00000010 _H | 32 | FFC12040 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR1_1 | 00000010 _H | 32 | FFC12044 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR1_2 | 00000010 _H | 32 | FFC12048 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR1_3 | 00000010 _H | 32 | FFC1204C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR1_4 | 00000010 _H | 32 | FFC12050 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR2_0 | 00000010 _H | 32 | FFC12080 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR2_1 | 00000010 _H | 32 | FFC12084 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR2_2 | 00000010 _H | 32 | FFC12088 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port control register | PCR2_3 | 00000010 _H | 32 | FFC1208C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (11/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------|---------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PORT | Port control register | PCR2_4 | 00000010 _H | 32 | FFC12090 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_5 | 00000010 _H | 32 | FFC12094 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_6 | 00000010 _H | 32 | FFC12098 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_7 | 00000010 _H | 32 | FFC1209C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_8 | 00000010 _H | 32 | FFC120A0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_9 | 00000010 _H | 32 | FFC120A4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_10 | 00000010 _H | 32 | FFC120A8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_11 | 00000010 _H | 32 | FFC120AC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_12 | 00000010 _H | 32 | FFC120B0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_13 | 00000010 _H | 32 | FFC120B4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_14 | 00000010 _H | 32 | FFC120B8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR2_15 | 00000010 _H | 32 | FFC120BC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_0 | 00000010 _H | 32 | FFC120C0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_1 | 00000010 _H | 32 | FFC120C4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_2 | 00000010 _H | 32 | FFC120C8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_3 | 00000010 _H | 32 | FFC120CC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_4 | 00000010 _H | 32 | FFC120D0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_5 | 00000010 _H | 32 | FFC120D4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_6 | 00000010 _H | 32 | FFC120D8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_7 | 00000010 _H | 32 | FFC120DC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_8 | 00000010 _H | 32 | FFC120E0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_9 | 00000010 _H | 32 | FFC120E4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_10 | 00000010 _H | 32 | FFC120E8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_11 | 00000010 _H | 32 | FFC120EC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_12 | 00000010 _H | 32 | FFC120F0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_13 | 00000010 _H | 32 | FFC120F4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR3_14 | 00000010 _H | 32 | FFC120F8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_0 | 00000010 _H | 32 | FFC12100 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_1 | 00000010 _H | 32 | FFC12104 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_2 | 00000010 _H | 32 | FFC12108 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_3 | 00000010 _H | 32 | FFC1210C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_4 | 00000010 _H | 32 | FFC12110 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_5 | 00000010 _H | 32 | FFC12114 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_6 | 00000010 _H | 32 | FFC12118 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_7 | 00000010 _H | 32 | FFC1211C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_8 | 00000010 _H | 32 | FFC12120 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_9 | 00000010 _H | 32 | FFC12124 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_10 | 00000010 _H | 32 | FFC12128 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_11 | 00000010 _H | 32 | FFC1212C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_12 | 00000010 _H | 32 | FFC12130 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_13 | 00000010 _H | 32 | FFC12134 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR4_14 | 00000010 _H | 32 | FFC12138 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_0 | 00000010 _H | 32 | FFC12140 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_1 | 00000010 _H | 32 | FFC12144 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_2 | 00000010 _H | 32 | FFC12148 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_3 | 00000010 _H | 32 | FFC1214C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_4 | 00000010 _H | 32 | FFC12150 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_5 | 00000010 _H | 32 | FFC12154 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (12/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|---------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PORT | Port control register | PCR5_6 | 00000010 _H | 32 | FFC12158 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_7 | 00000010 _H | 32 | FFC1215C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_8 | 00000010 _H | 32 | FFC12160 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_9 | 00000010 _H | 32 | FFC12164 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_10 | 00000010 _H | 32 | FFC12168 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_11 | 00000010 _H | 32 | FFC1216C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_12 | 00000010 _H | 32 | FFC12170 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_13 | 00000010 _H | 32 | FFC12174 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_14 | 00000010 _H | 32 | FFC12178 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port control register | PCR5_15 | 00000010 _H | 32 | FFC1217C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port input buffer control register | PIBC0 | 0000 _H | 16 | FFC14000 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port bi-direction control register | PBDC0 | 0000 _H | 16 | FFC14004 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port IP control register | PIPC0 | 0000 _H | 16 | FFC14008 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-up option register | PU0 | 0000 _H | 16 | FFC1400C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-down option register | PD0 | 0000 _H | 16 | FFC14010 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port open-drain control register | PODC0 | 00000400 _H | 32 | FFC14014 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port drive strength control register | PDSC0 | 00000400 _H | 32 | FFC14018 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port universal control register | PUCC0 | 00000000 _H | 32 | FFC14028 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port input buffer selection register | PISA0 | 0400 _H | 16 | FFC1402C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port open-drain control expansion register | PODCE0 | 00000000 _H | 32 | FFC1403C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port input buffer control register | PIBC1 | 0000 _H | 16 | FFC14040 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port bi-direction control register | PBDC1 | 0000 _H | 16 | FFC14044 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port IP control register | PIPC1 | 0000 _H | 16 | FFC14048 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-up option register | PU1 | 0000 _H | 16 | FFC1404C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-down option register | PD1 | 0000 _H | 16 | FFC14050 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port open-drain control register | PODC1 | 00000000 _H | 32 | FFC14054 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port drive strength control register | PDSC1 | 00000000 _H | 32 | FFC14058 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port universal control register | PUCC1 | 00000000 _H | 32 | FFC14068 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port input buffer selection register | PISA1 | 0000 _H | 16 | FFC1406C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port open-drain control expansion register | PODCE1 | 00000000 _H | 32 | FFC1407C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port input buffer control register | PIBC2 | 0000 _H | 16 | FFC14080 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port bi-direction control register | PBDC2 | 0000 _H | 16 | FFC14084 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port IP control register | PIPC2 | 0000 _H | 16 | FFC14088 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-up option register | PU2 | 0000 _H | 16 | FFC1408C _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-down option register | PD2 | 0000 _H | 16 | FFC14090 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port open-drain control register | PODC2 | 00000000 _H | 32 | FFC14094 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port drive strength control register | PDSC2 | 00000000 _H | 32 | FFC14098 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port universal control register | PUCC2 | 00000000 _H | 32 | FFC140A8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port input buffer selection register | PISA2 | 0000 _H | 16 | FFC140AC _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port open-drain control expansion register | PODCE2 | 00000000 _H | 32 | FFC140BC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port input buffer control register | PIBC3 | 0000 _H | 16 | FFC140C0 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port bi-direction control register | PBDC3 | 0000 _H | 16 | FFC140C4 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port IP control register | PIPC3 | 0000 _H | 16 | FFC140C8 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-up option register | PU3 | 0000 _H | 16 | FFC140CC _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Pull-down option register | PD3 | 0000 _H | 16 | FFC140D0 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port open-drain control register | PODC3 | 00000000 _H | 32 | FFC140D4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port drive strength control register | PDSC3 | 00000000 _H | 32 | FFC140D8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PORT | Port universal control register | PUCC3 | 00000000 _H | 32 | FFC140E8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (13/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PORT | Port input buffer selection register | PISA3 | 0000 _H | 16 | FFC140EC _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port open-drain control expansion register | PODCE3 | 00000000 _H | 32 | FFC140FC _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port input buffer control register | PIBC4 | 0000 _H | 16 | FFC14100 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port bi-direction control register | PBDC4 | 0000 _H | 16 | FFC14104 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port IP control register | PIPC4 | 0000 _H | 16 | FFC14108 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Pull-up option register | PU4 | 0000 _H | 16 | FFC1410C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Pull-down option register | PD4 | 0000 _H | 16 | FFC14110 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port open-drain control register | PODC4 | 00000000 _H | 32 | FFC14114 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port drive strength control register | PDSC4 | 00000000 _H | 32 | FFC14118 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port universal control register | PUCC4 | 00000000 _H | 32 | FFC14128 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port input buffer selection register | PISA4 | 0000 _H | 16 | FFC1412C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port open-drain control expansion register | PODCE4 | 00000000 _H | 32 | FFC1413C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port input buffer control register | PIBC5 | 0000 _H | 16 | FFC14140 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port bi-direction control register | PBDC5 | 0000 _H | 16 | FFC14144 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port IP control register | PIPC5 | 0000 _H | 16 | FFC14148 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Pull-up option register | PU5 | 0000 _H | 16 | FFC1414C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Pull-down option register | PD5 | 0000 _H | 16 | FFC14150 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| PORT | Port open-drain control register | PODC5 | 00000000 _H | 32 | FFC14154 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port drive strength control register | PDSC5 | 00000000 _H | 32 | FFC14158 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port universal control register | PUCC5 | 00000000 _H | 32 | FFC14168 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORT | Port open-drain control expansion register | PODCE5 | 00000000 _H | 32 | FFC1417C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port register | JP0 | 00 _H | 8 | FFC20000 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port set/reset register | JPSR0 | 00000000 _H | 32 | FFC20004 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port NOT register | JPNOT0 | 00 _H | 8 | FFC20008 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port pin read register | JPPR0 | 00 _H | 8 | FFC2000C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port mode register | JPM0 | FF _H | 8 | FFC20010 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port mode control register | JPMC0 | 00 _H | 8 | FFC20014 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port function control expansion register | JPFCE0 | 00 _H | 8 | FFC2001C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port mode set/reset register | JPMSR0 | 000000FF _H | 32 | FFC20020 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port mode control set/reset register | JPMCSR0 | 00000000 _H | 32 | FFC20024 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port control register | JPCR0_0 | 00000010 _H | 32 | FFC22000 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port control register | JPCR0_1 | 00000010 _H | 32 | FFC22004 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port control register | JPCR0_2 | 00000010 _H | 32 | FFC22008 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port control register | JPCR0_3 | 00000010 _H | 32 | FFC2200C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port control register | JPCR0_4 | 00000010 _H | 32 | FFC22010 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port control register | JPCR0_5 | 00000010 _H | 32 | FFC22014 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port input buffer control register | JPIBC0 | 00 _H | 8 | FFC24000 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port bi-direction control register | JPBDC0 | 00 _H | 8 | FFC24004 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Pull-up option register | JPU0 | 00 _H | 8 | FFC2400C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Pull-down option register | JPD0 | 00 _H | 8 | FFC24010 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port open-drain control register | JPODC0 | 00000000 _H | 32 | FFC24014 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port drive strength control register | JPDSC0 | 00000000 _H | 32 | FFC24018 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port universal control register | JPUCC0 | 00000000 _H | 32 | FFC24028 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PORTJ | Port input buffer selection register | JPISA0 | 00 _H | 8 | FFC2402C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA0 | Digital noise elimination control register | DNFA0CTL | 00 _H | 8 | FFC30000 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA0 | Digital noise elimination enable register | DNFA0EN | 0000 _H | 16 | FFC30004 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA0 | Digital noise elimination enable register | DNFA0ENL | 00 _H | 8 | FFC3000C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA1 | Digital noise elimination control register | DNFA1CTL | 00 _H | 8 | FFC30100 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (14/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-----------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DNFA1 | Digital noise elimination enable register | DNFA1EN | 0000 _H | 16 | FFC30104 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA1 | Digital noise elimination enable register | DNFA1ENL | 00 _H | 8 | FFC3010C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA2 | Digital noise elimination control register | DNFA2CTL | 00 _H | 8 | FFC30200 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA2 | Digital noise elimination enable register | DNFA2EN | 0000 _H | 16 | FFC30204 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA2 | Digital noise elimination enable register | DNFA2ENL | 00 _H | 8 | FFC3020C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA3 | Digital noise elimination control register | DNFA3CTL | 00 _H | 8 | FFC30300 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA3 | Digital noise elimination enable register | DNFA3EN | 0000 _H | 16 | FFC30304 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA3 | Digital noise elimination enable register | DNFA3ENL | 00 _H | 8 | FFC3030C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA4 | Digital noise elimination control register | DNFA4CTL | 00 _H | 8 | FFC30400 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA4 | Digital noise elimination enable register | DNFA4EN | 0000 _H | 16 | FFC30404 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA4 | Digital noise elimination enable register | DNFA4ENL | 00 _H | 8 | FFC3040C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA5 | Digital noise elimination control register | DNFA5CTL | 00 _H | 8 | FFC30500 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA5 | Digital noise elimination enable register | DNFA5EN | 0000 _H | 16 | FFC30504 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA5 | Digital noise elimination enable register | DNFA5ENL | 00 _H | 8 | FFC3050C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA6 | Digital noise elimination control register | DNFA6CTL | 00 _H | 8 | FFC30600 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA6 | Digital noise elimination enable register | DNFA6EN | 0000 _H | 16 | FFC30604 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA6 | Digital noise elimination enable register | DNFA6ENL | 00 _H | 8 | FFC3060C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA7 | Digital noise elimination control register | DNFA7CTL | 00 _H | 8 | FFC30700 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA7 | Digital noise elimination enable register | DNFA7EN | 0000 _H | 16 | FFC30704 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA7 | Digital noise elimination enable register | DNFA7ENL | 00 _H | 8 | FFC3070C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA8 | Digital noise elimination control register | DNFA8CTL | 00 _H | 8 | FFC30800 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA8 | Digital noise elimination enable register | DNFA8EN | 0000 _H | 16 | FFC30804 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA8 | Digital noise elimination enable register | DNFA8ENL | 00 _H | 8 | FFC3080C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA9 | Digital noise elimination control register | DNFA9CTL | 00 _H | 8 | FFC30900 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA9 | Digital noise elimination enable register | DNFA9EN | 0000 _H | 16 | FFC30904 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA9 | Digital noise elimination enable register | DNFA9ENL | 00 _H | 8 | FFC3090C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA10 | Digital noise elimination control register | DNFA10CTL | 00 _H | 8 | FFC30A00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA10 | Digital noise elimination enable register | DNFA10EN | 0000 _H | 16 | FFC30A04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA10 | Digital noise elimination enable register | DNFA10ENL | 00 _H | 8 | FFC30A0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA11 | Digital noise elimination control register | DNFA11CTL | 00 _H | 8 | FFC30B00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA11 | Digital noise elimination enable register | DNFA11EN | 0000 _H | 16 | FFC30B04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA11 | Digital noise elimination enable register | DNFA11ENL | 00 _H | 8 | FFC30B0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA12 | Digital noise elimination control register | DNFA12CTL | 00 _H | 8 | FFC30C00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA12 | Digital noise elimination enable register | DNFA12EN | 0000 _H | 16 | FFC30C04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA12 | Digital noise elimination enable register | DNFA12ENL | 00 _H | 8 | FFC30C0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA13 | Digital noise elimination control register | DNFA13CTL | 00 _H | 8 | FFC30D00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA13 | Digital noise elimination enable register | DNFA13EN | 0000 _H | 16 | FFC30D04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA13 | Digital noise elimination enable register | DNFA13ENL | 00 _H | 8 | FFC30D0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA14 | Digital noise elimination control register | DNFA14CTL | 00 _H | 8 | FFC30E00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA14 | Digital noise elimination enable register | DNFA14EN | 0000 _H | 16 | FFC30E04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA14 | Digital noise elimination enable register | DNFA14ENL | 00 _H | 8 | FFC30E0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA15 | Digital noise elimination control register | DNFA15CTL | 00 _H | 8 | FFC30F00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA15 | Digital noise elimination enable register | DNFA15EN | 0000 _H | 16 | FFC30F04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA15 | Digital noise elimination enable register | DNFA15ENL | 00 _H | 8 | FFC30F0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA16 | Digital noise elimination control register | DNFA16CTL | 00 _H | 8 | FFC31000 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA16 | Digital noise elimination enable register | DNFA16EN | 0000 _H | 16 | FFC31004 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA16 | Digital noise elimination enable register | DNFA16ENL | 00 _H | 8 | FFC3100C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA17 | Digital noise elimination control register | DNFA17CTL | 00 _H | 8 | FFC31100 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (15/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-----------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DNFA17 | Digital noise elimination enable register | DNFA17EN | 0000 _H | 16 | FFC31104 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA17 | Digital noise elimination enable register | DNFA17ENL | 00 _H | 8 | FFC3110C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA18 | Digital noise elimination control register | DNFA18CTL | 00 _H | 8 | FFC31200 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA18 | Digital noise elimination enable register | DNFA18EN | 0000 _H | 16 | FFC31204 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA18 | Digital noise elimination enable register | DNFA18ENL | 00 _H | 8 | FFC3120C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA19 | Digital noise elimination control register | DNFA19CTL | 00 _H | 8 | FFC31300 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA19 | Digital noise elimination enable register | DNFA19EN | 0000 _H | 16 | FFC31304 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA19 | Digital noise elimination enable register | DNFA19ENL | 00 _H | 8 | FFC3130C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA20 | Digital noise elimination control register | DNFA20CTL | 00 _H | 8 | FFC31400 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA20 | Digital noise elimination enable register | DNFA20EN | 0000 _H | 16 | FFC31404 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA20 | Digital noise elimination enable register | DNFA20ENL | 00 _H | 8 | FFC3140C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA21 | Digital noise elimination control register | DNFA21CTL | 00 _H | 8 | FFC31500 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA21 | Digital noise elimination enable register | DNFA21EN | 0000 _H | 16 | FFC31504 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA21 | Digital noise elimination enable register | DNFA21ENL | 00 _H | 8 | FFC3150C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA22 | Digital noise elimination control register | DNFA22CTL | 00 _H | 8 | FFC31600 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA22 | Digital noise elimination enable register | DNFA22EN | 0000 _H | 16 | FFC31604 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA22 | Digital noise elimination enable register | DNFA22ENL | 00 _H | 8 | FFC3160C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA23 | Digital noise elimination control register | DNFA23CTL | 00 _H | 8 | FFC31700 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA23 | Digital noise elimination enable register | DNFA23EN | 0000 _H | 16 | FFC31704 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA23 | Digital noise elimination enable register | DNFA23ENL | 00 _H | 8 | FFC3170C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA24 | Digital noise elimination control register | DNFA24CTL | 00 _H | 8 | FFC31800 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA24 | Digital noise elimination enable register | DNFA24EN | 0000 _H | 16 | FFC31804 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA24 | Digital noise elimination enable register | DNFA24ENL | 00 _H | 8 | FFC3180C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA25 | Digital noise elimination control register | DNFA25CTL | 00 _H | 8 | FFC31900 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA25 | Digital noise elimination enable register | DNFA25EN | 0000 _H | 16 | FFC31904 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA25 | Digital noise elimination enable register | DNFA25ENL | 00 _H | 8 | FFC3190C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA26 | Digital noise elimination control register | DNFA26CTL | 00 _H | 8 | FFC31A00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA26 | Digital noise elimination enable register | DNFA26EN | 0000 _H | 16 | FFC31A04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA26 | Digital noise elimination enable register | DNFA26ENL | 00 _H | 8 | FFC31A0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA27 | Digital noise elimination control register | DNFA27CTL | 00 _H | 8 | FFC31B00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA27 | Digital noise elimination enable register | DNFA27EN | 0000 _H | 16 | FFC31B04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA27 | Digital noise elimination enable register | DNFA27ENL | 00 _H | 8 | FFC31B0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA28 | Digital noise elimination control register | DNFA28CTL | 00 _H | 8 | FFC31C00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA28 | Digital noise elimination enable register | DNFA28EN | 0000 _H | 16 | FFC31C04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA28 | Digital noise elimination enable register | DNFA28ENL | 00 _H | 8 | FFC31C0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA29 | Digital noise elimination control register | DNFA29CTL | 00 _H | 8 | FFC31D00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA29 | Digital noise elimination enable register | DNFA29EN | 0000 _H | 16 | FFC31D04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA29 | Digital noise elimination enable register | DNFA29ENL | 00 _H | 8 | FFC31D0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA30 | Digital noise elimination control register | DNFA30CTL | 00 _H | 8 | FFC31E00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA30 | Digital noise elimination enable register | DNFA30EN | 0000 _H | 16 | FFC31E04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA30 | Digital noise elimination enable register | DNFA30ENL | 00 _H | 8 | FFC31E0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA31 | Digital noise elimination control register | DNFA31CTL | 00 _H | 8 | FFC31F00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA31 | Digital noise elimination enable register | DNFA31EN | 0000 _H | 16 | FFC31F04 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA31 | Digital noise elimination enable register | DNFA31ENL | 00 _H | 8 | FFC31F0C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA32 | Digital noise elimination control register | DNFA32CTL | 00 _H | 8 | FFC32000 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA32 | Digital noise elimination enable register | DNFA32EN | 0000 _H | 16 | FFC32004 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA32 | Digital noise elimination enable register | DNFA32ENL | 00 _H | 8 | FFC3200C _H | 5 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DNFA33 | Digital noise elimination control register | DNFA33CTL | 00 _H | 8 | FFC32100 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (16/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-----------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DNFA33 | Digital noise elimination enable register | DNFA33EN | 0000 _H | 16 | FFC32104 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA33 | Digital noise elimination enable register | DNFA33ENL | 00 _H | 8 | FFC3210C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA34 | Digital noise elimination control register | DNFA34CTL | 00 _H | 8 | FFC32200 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA34 | Digital noise elimination enable register | DNFA34EN | 0000 _H | 16 | FFC32204 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA34 | Digital noise elimination enable register | DNFA34ENL | 00 _H | 8 | FFC3220C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA35 | Digital noise elimination control register | DNFA35CTL | 00 _H | 8 | FFC32300 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA35 | Digital noise elimination enable register | DNFA35EN | 0000 _H | 16 | FFC32304 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA35 | Digital noise elimination enable register | DNFA35ENL | 00 _H | 8 | FFC3230C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA36 | Digital noise elimination control register | DNFA36CTL | 00 _H | 8 | FFC32400 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA36 | Digital noise elimination enable register | DNFA36EN | 0000 _H | 16 | FFC32404 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA36 | Digital noise elimination enable register | DNFA36ENL | 00 _H | 8 | FFC3240C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA37 | Digital noise elimination control register | DNFA37CTL | 00 _H | 8 | FFC32500 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA37 | Digital noise elimination enable register | DNFA37EN | 0000 _H | 16 | FFC32504 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA37 | Digital noise elimination enable register | DNFA37ENL | 00 _H | 8 | FFC3250C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA38 | Digital noise elimination control register | DNFA38CTL | 00 _H | 8 | FFC32600 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA38 | Digital noise elimination enable register | DNFA38EN | 0000 _H | 16 | FFC32604 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA38 | Digital noise elimination enable register | DNFA38ENL | 00 _H | 8 | FFC3260C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA39 | Digital noise elimination control register | DNFA39CTL | 00 _H | 8 | FFC32700 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA39 | Digital noise elimination enable register | DNFA39EN | 0000 _H | 16 | FFC32704 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA39 | Digital noise elimination enable register | DNFA39ENL | 00 _H | 8 | FFC3270C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA40 | Digital noise elimination control register | DNFA40CTL | 00 _H | 8 | FFC32800 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA40 | Digital noise elimination enable register | DNFA40EN | 0000 _H | 16 | FFC32804 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA40 | Digital noise elimination enable register | DNFA40ENL | 00 _H | 8 | FFC3280C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA41 | Digital noise elimination control register | DNFA41CTL | 00 _H | 8 | FFC32900 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA41 | Digital noise elimination enable register | DNFA41EN | 0000 _H | 16 | FFC32904 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA41 | Digital noise elimination enable register | DNFA41ENL | 00 _H | 8 | FFC3290C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA42 | Digital noise elimination control register | DNFA42CTL | 00 _H | 8 | FFC32A00 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA42 | Digital noise elimination enable register | DNFA42EN | 0000 _H | 16 | FFC32A04 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA42 | Digital noise elimination enable register | DNFA42ENL | 00 _H | 8 | FFC32A0C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| DNFA43 | Digital noise elimination control register | DNFA43CTL | 00 _H | 8 | FFC32B00 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| DNFA43 | Digital noise elimination enable register | DNFA43EN | 0000 _H | 16 | FFC32B04 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| DNFA43 | Digital noise elimination enable register | DNFA43ENL | 00 _H | 8 | FFC32B0C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA0 | Filter control register | FCLA0CTL0 | 00 _H | 8 | FFC34000 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL0 | 00 _H | 8 | FFC34020 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL1 | 00 _H | 8 | FFC34024 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL2 | 00 _H | 8 | FFC34028 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL3 | 00 _H | 8 | FFC3402C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL4 | 00 _H | 8 | FFC34030 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL5 | 00 _H | 8 | FFC34034 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL6 | 00 _H | 8 | FFC34038 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA1 | Filter control register | FCLA1CTL7 | 00 _H | 8 | FFC3403C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA2 | Filter control register | FCLA2CTL0 | 00 _H | 8 | FFC34040 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA2 | Filter control register | FCLA2CTL1 | 00 _H | 8 | FFC34044 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA2 | Filter control register | FCLA2CTL2 | 00 _H | 8 | FFC34048 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA2 | Filter control register | FCLA2CTL3 | 00 _H | 8 | FFC3404C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA2 | Filter control register | FCLA2CTL4 | 00 _H | 8 | FFC34050 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA3 | Filter control register | FCLA3CTL0 | 00 _H | 8 | FFC34060 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA3 | Filter control register | FCLA3CTL1 | 00 _H | 8 | FFC34064 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (17/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| FCLA3 | Filter control register | FCLA3CTL2 | 00 _H | 8 | FFC34068 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA3 | Filter control register | FCLA3CTL3 | 00 _H | 8 | FFC3406C _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA3 | Filter control register | FCLA3CTL4 | 00 _H | 8 | FFC34070 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA3 | Filter control register | FCLA3CTL5 | 00 _H | 8 | FFC34074 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA4 | Filter control register | FCLA4CTL0 | 00 _H | 8 | FFC34080 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| FCLA4 | Filter control register | FCLA4CTL1 | 00 _H | 8 | FFC34084 _H | 5 | 1, 8 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT00 | 0605FE1B _H | 32 | FFC40000 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID00 | FFFFFFF _H | 32 | FFC40004 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT01 | 0605FE17 _H | 32 | FFC40008 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID01 | FFFFFFF _H | 32 | FFC4000C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT02 | 0605FE1B _H | 32 | FFC40010 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID02 | FFFFFFF _H | 32 | FFC40014 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT03 | 0605FE17 _H | 32 | FFC40018 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID03 | FFFFFFF _H | 32 | FFC4001C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT04 | 0605FE1B _H | 32 | FFC40020 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID04 | FFFFFFF _H | 32 | FFC40024 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT05 | 0605FE17 _H | 32 | FFC40028 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID05 | FFFFFFF _H | 32 | FFC4002C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT06 | 0605FE1B _H | 32 | FFC40030 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID06 | FFFFFFF _H | 32 | FFC40034 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT07 | 0605FE17 _H | 32 | FFC40038 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID07 | FFFFFFF _H | 32 | FFC4003C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT08 | 0605FE1B _H | 32 | FFC40040 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID08 | FFFFFFF _H | 32 | FFC40044 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT09 | 0605FE17 _H | 32 | FFC40048 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID09 | FFFFFFF _H | 32 | FFC4004C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT10 | 0605FE1B _H | 32 | FFC40050 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID10 | FFFFFFF _H | 32 | FFC40054 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT11 | 0605FE17 _H | 32 | FFC40058 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID11 | FFFFFFF _H | 32 | FFC4005C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT12 | 0605FE1B _H | 32 | FFC40060 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID12 | FFFFFFF _H | 32 | FFC40064 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT13 | 0605FE17 _H | 32 | FFC40068 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID13 | FFFFFFF _H | 32 | FFC4006C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT14 | 0605FE1B _H | 32 | FFC40070 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID14 | FFFFFFF _H | 32 | FFC40074 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5APROT15 | 0605FE17 _H | 32 | FFC40078 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5ASPID15 | FFFFFFF _H | 32 | FFC4007C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT00 | 0605FE1B _H | 32 | FFC40080 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID00 | FFFFFFF _H | 32 | FFC40084 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT01 | 0605FE17 _H | 32 | FFC40088 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID01 | FFFFFFF _H | 32 | FFC4008C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT02 | 0605FE1B _H | 32 | FFC40090 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID02 | FFFFFFF _H | 32 | FFC40094 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT03 | 0605FE17 _H | 32 | FFC40098 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID03 | FFFFFFF _H | 32 | FFC4009C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT06 | 0605FE1B _H | 32 | FFC400B0 _H | 5 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID06 | FFFFFFF _H | 32 | FFC400B4 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (18/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT07 | 0605FE17 _H | 32 | FFC400B8 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID07 | FFFFFFF _H | 32 | FFC400BC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT08 | 0605FE1B _H | 32 | FFC400C0 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID08 | FFFFFFF _H | 32 | FFC400C4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT09 | 0605FE17 _H | 32 | FFC400C8 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID09 | FFFFFFF _H | 32 | FFC400CC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT10 | 0605FE1B _H | 32 | FFC400D0 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID10 | FFFFFFF _H | 32 | FFC400D4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT11 | 0605FE17 _H | 32 | FFC400D8 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID11 | FFFFFFF _H | 32 | FFC400DC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT12 | 0605FE1B _H | 32 | FFC400E0 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID12 | FFFFFFF _H | 32 | FFC400E4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT13 | 0605FE17 _H | 32 | FFC400E8 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID13 | FFFFFFF _H | 32 | FFC400EC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT14 | 0605FE1B _H | 32 | FFC400F0 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID14 | FFFFFFF _H | 32 | FFC400F4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5BPROT15 | 0605FE17 _H | 32 | FFC400F8 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5BSPID15 | FFFFFFF _H | 32 | FFC400FC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5CPROT00 | 0605FE1B _H | 32 | FFC40100 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5CSPID00 | FFFFFFF _H | 32 | FFC40104 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5CPROT01 | 0605FE17 _H | 32 | FFC40108 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5CSPID01 | FFFFFFF _H | 32 | FFC4010C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5CPROT02 | 0605FE1B _H | 32 | FFC40110 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5CSPID02 | FFFFFFF _H | 32 | FFC40114 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5CPROT03 | 0605FE17 _H | 32 | FFC40118 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5CSPID03 | FFFFFFF _H | 32 | FFC4011C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5CPROT04 | 0605FE1B _H | 32 | FFC40120 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5CSPID04 | FFFFFFF _H | 32 | FFC40124 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-Bus Guard Protection Setting Register | FSGD5CPROT05 | 0605FE17 _H | 32 | FFC40128 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard SPID Setting Register | FSGD5CSPID05 | FFFFFFF _H | 32 | FFC4012C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard ERRSLV Control Register | ERRSLV5CTL | 00 _H | 8 | FFC40400 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard ERRSLV Status Register | ERRSLV5STAT | 00000000 _H | 32 | FFC40404 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG5 | P-bus Guard ERRSLV Error Transfer Type Register | ERRSLV5TYPE | 00000000 _H | 32 | FFC4040C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC5 | P-Bus Data and Address ECC Control Register | APEC2ECCCTL | 00000000 _H | 32 | FFC40800 _H | 5 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC5 | P-Bus Error Information Control Register | APEC2ERRINT | 00000073 _H | 32 | FFC40804 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC5 | P-Bus Data and Address ECC SED/DED Status Clear Register | APEC2STCLR | 00000000 _H | 32 | FFC40808 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC5 | P-Bus Data and Address Error Count Overflow Status Register | APEC2OVFSTR | 00000000 _H | 32 | FFC4080C _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC5 | P-Bus Data and Address ECC SED/DED Status Register | APEC21STERSTR | 00000000 _H | 32 | FFC40810 _H | 5 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC5 | P-Bus Data and Address ECC SED/DED Address Register | APEC21STEADR0 | 00000000 _H | 32 | FFC40850 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GRG | Global-RAM FS Guard Protection Setting Register0 | MGDGRPROT0 | 07FFFE10 _H | 32 | FFC49000 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GRG | Global-RAM FS Guard SPID Setting Register0 | MGDGRSPID0 | FFFFFFF _H | 32 | FFC49004 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GRG | Global-RAM FS Guard Base Address Register0 | MGDGRBAD0 | 00000000 _H | 32 | FFC49008 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GRG | Global-RAM FS Guard Address Valid Register0 | MGDGRADV0 | 00000000 _H | 32 | FFC4900C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GRG | Global-RAM FS Guard Protection Setting Register1 | MGDGRPROT1 | 07FFFE10 _H | 32 | FFC49010 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GRG | Global-RAM FS Guard SPID Setting Register1 | MGDGRSPID1 | FFFFFFF _H | 32 | FFC49014 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (19/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| GRG | Global-RAM FS Guard Base Address Register1 | MGDGRBAD1 | 00000000 _H | 32 | FFC49018 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Address Valid Register1 | MGDGRADV1 | 00000000 _H | 32 | FFC4901C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Protection Setting Register2 | MGDGRPROT2 | 07FFFE10 _H | 32 | FFC49020 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard SPID Setting Register2 | MGDGRSPID2 | FFFFFFF _H | 32 | FFC49024 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Base Address Register2 | MGDGRBAD2 | 00000000 _H | 32 | FFC49028 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Address Valid Register2 | MGDGRADV2 | 00000000 _H | 32 | FFC4902C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Protection Setting Register3 | MGDGRPROT3 | 07FFFE10 _H | 32 | FFC49030 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard SPID Setting Register3 | MGDGRSPID3 | FFFFFFF _H | 32 | FFC49034 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Base Address Register3 | MGDGRBAD3 | 00000000 _H | 32 | FFC49038 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Address Valid Register3 | MGDGRADV3 | 00000000 _H | 32 | FFC4903C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Protection Setting Register4 | MGDGRPROT4 | 07FFFE10 _H | 32 | FFC49040 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard SPID Setting Register4 | MGDGRSPID4 | FFFFFFF _H | 32 | FFC49044 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Base Address Register3 | MGDGRBAD4 | 00000000 _H | 32 | FFC49048 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Address Valid Register4 | MGDGRADV4 | 00000000 _H | 32 | FFC4904C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Protection Setting Register5 | MGDGRPROT5 | 07FFFE10 _H | 32 | FFC49050 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard SPID Setting Register5 | MGDGRSPID5 | FFFFFFF _H | 32 | FFC49054 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Base Address Register5 | MGDGRBAD5 | 00000000 _H | 32 | FFC49058 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Address Valid Register5 | MGDGRADV5 | 00000000 _H | 32 | FFC4905C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Protection Setting Register6 | MGDGRPROT6 | 07FFFE10 _H | 32 | FFC49060 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard SPID Setting Register6 | MGDGRSPID6 | FFFFFFF _H | 32 | FFC49064 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Base Address Register6 | MGDGRBAD6 | 00000000 _H | 32 | FFC49068 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Address Valid Register6 | MGDGRADV6 | 00000000 _H | 32 | FFC4906C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Protection Setting Register7 | MGDGRPROT7 | 07FFFE10 _H | 32 | FFC49070 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard SPID Setting Register7 | MGDGRSPID7 | FFFFFFF _H | 32 | FFC49074 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Base Address Register7 | MGDGRBAD7 | 00000000 _H | 32 | FFC49078 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Address Valid Register7 | MGDGRADV7 | 00000000 _H | 32 | FFC4907C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Control Register(VCI2GRAM) | MGDGRSCTL_VCI2GRAM | 00000000 _H | 32 | FFC49100 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Error Status Register (VCI2GRAM) | MGDGRSSTAT_VCI2GRAM | 00000000 _H | 32 | FFC49104 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Error Access Type Register(VCI2GRAM) | MGDGRSTYPE_VCI2GRAM | XXXXXXXX _H | 32 | FFC4910C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Control Register (PE1) | MGDGRSCTL_PE1 | 00000000 _H | 32 | FFC49200 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Error Status Register (PE1) | MGDGRSSTAT_PE1 | 00000000 _H | 32 | FFC49204 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Error Access Type Register (PE1) | MGDGRSTYPE_PE1 | XXXXXXXX _H | 32 | FFC4920C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Control Register (AXI2GRAM) | MGDGRSCTL_AXI2GRAM | 00000000 _H | 32 | FFC49700 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Error Status Register (AXI2GRAM) | MGDGRSSTAT_AXI2GRAM | 00000000 _H | 32 | FFC49704 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRG | Global-RAM FS Guard Error Access Type Register (AXI2GRAM) | MGDGRSTYPE_AXI2GRAM | XXXXXXXX _H | 32 | FFC4970C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Error Status Control Register | PGERRSTATCTL_PE1 | 00000000 _H | 32 | FFC4A200 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Error Status Register | PGERRSTAT_PE1 | 00000000 _H | 32 | FFC4A204 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Error Information Register | PGERRINFO_PE1 | 00000000 _H | 32 | FFC4A208 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_PDMACM_A | 060DFE1B _H | 32 | FFC4C000 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_PDMACM_A | FFFFFFF _H | 32 | FFC4C004 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_PDMACM_B | 060DFE17 _H | 32 | FFC4C008 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_PDMACM_B | FFFFFFF _H | 32 | FFC4C00C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_PDMACH_A | 060DFE1B _H | 32 | FFC4C010 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (20/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | | | |
|----------|--|----------------------|--|----------------|---------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|---|---|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM | | |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_PDMACH_A | FFFFFFFH | H | 32 | FFC4C014 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_PDMACH_B | 060DFE17 | H | 32 | FFC4C018 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_PDMACH_B | FFFFFFFH | H | 32 | FFC4C01C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_INT2_A | 060DFE1B | H | 32 | FFC4C020 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_INT2_A | FFFFFFFH | H | 32 | FFC4C024 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_INT2_B | 060DFE17 | H | 32 | FFC4C028 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_INT2_B | FFFFFFFH | H | 32 | FFC4C02C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_PBG_A | 064DFE1B | H | 32 | FFC4C040 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_PBG_A | FFFFFFFH | H | 32 | FFC4C044 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_PBG_B | 064DFE17 | H | 32 | FFC4C048 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_PBG_B | FFFFFFFH | H | 32 | FFC4C04C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_SP1_A | 060DFE1B | H | 32 | FFC4C050 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_SP1_A | FFFFFFFH | H | 32 | FFC4C054 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_SP1_B | 060DFE17 | H | 32 | FFC4C058 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_SP1_B | FFFFFFFH | H | 32 | FFC4C05C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_SP4_A | 060DFE1B | H | 32 | FFC4C070 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_SP4_A | FFFFFFFH | H | 32 | FFC4C074 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard Protection Setting Register | APBFSGDPROT_SP4_B | 060DFE17 | H | 32 | FFC4C078 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus FS Guard SPID Setting Register | APBFSGDSPID_SP4_B | FFFFFFFH | H | 32 | FFC4C07C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | ERRSLV Control Register for PFSS P-bus FS Guard Slave0 | ERRSLVCTL_PFSS0 | 00000000 | H | 32 | FFC4C800 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | ERRSLV Status Register for PFSS P-bus FS Guard Slave0 | ERRSLVSTAT_PFSS0 | 00000000 | H | 32 | FFC4C804 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | ERRSLV Error Transfer Type Register for PFSS P-bus FS Guard Slave0 | ERRSLVTYPE_PFSS0 | 00000000 | H | 32 | FFC4C80C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | ERRSLV Control Register for PFSS P-bus FS Guard Slave1 | ERRSLVCTL_PFSS1 | 00000000 | H | 32 | FFC4C810 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | ERRSLV Status Register for PFSS P-bus FS Guard Slave1 | ERRSLVSTAT_PFSS1 | 00000000 | H | 32 | FFC4C814 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | ERRSLV Error Transfer Type Register for PFSS P-bus FS Guard Slave1 | ERRSLVTYPE_PFSS1 | 00000000 | H | 32 | FFC4C81C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PDMACOMP | PDMA Comparator Error Injection Control Register | PDMA_COMP_CNTRL | 00000000 | H | 32 | FFC4CA00 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FACI | BFA Selection Register | BFASCLR | This register is set to "1" when booted in serial programming mode, and cleared to "0" when booted in normal operating mode. | | 8 | FFC59008 | H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| FLASH | Data flash memory read cycle setting register | EEPRDCYCL | 0F | H | 8 | FFC59810 | H | 0 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT00 | 0605FE1B | H | 32 | FFC5A000 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID00 | FFFFFFFH | H | 32 | FFC5A004 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT01 | 0605FE17 | H | 32 | FFC5A008 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID01 | FFFFFFFH | H | 32 | FFC5A00C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT04 | 0605FE1B | H | 32 | FFC5A020 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID04 | FFFFFFFH | H | 32 | FFC5A024 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT05 | 0605FE17 | H | 32 | FFC5A028 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID05 | FFFFFFFH | H | 32 | FFC5A02C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT06 | 0605FE1B | H | 32 | FFC5A030 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID06 | FFFFFFFH | H | 32 | FFC5A034 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT07 | 0605FE17 | H | 32 | FFC5A038 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID07 | FFFFFFFH | H | 32 | FFC5A03C | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT10 | 0605FE1B | H | 32 | FFC5A050 | H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID10 | FFFFFFFH | H | 32 | FFC5A054 | H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (21/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBG0 | P-Bus Guard Protection Setting Register | FSGD0PROT11 | 0605FE17 _H | 32 | FFC5A058 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus Guard SPID Setting Register | FSGD0SPID11 | FFFFFFF _H | 32 | FFC5A05C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus Guard ERRSLV Control Register | ERRSLV0CTL | 00 _H | 8 | FFC5A100 _H | 0 | 8 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus Guard ERRSLV Status Register | ERRSLV0STAT | 00000000 _H | 32 | FFC5A104 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| PBG0 | P-bus Guard ERRSLV Error Transfer Type Register | ERRSLV0TYPE | 00000000 _H | 32 | FFC5A10C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data Flash ECC control register | DFECCCTL | 00000000 _H | 32 | FFC5B000 _H | 0 | 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data Flash error status register | DFERSTR | 00000000 _H | 32 | FFC5B004 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data Flash error status clear register | DFERSTC | 00000000 _H | 32 | FFC5B008 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data Flash error overflow status register | DFOVFSTR | 00000000 _H | 32 | FFC5B00C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data Flash error overflow status clear register | DFOVFSTC | 00000000 _H | 32 | FFC5B010 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data Flash error notification control register | DFERRINT | 00000006 _H | 32 | FFC5B014 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data Flash 1st error address register | DFEADR | 00000000 _H | 32 | FFC5B018 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCEEP | Data flash test control register | DFTSTCTL | 00000000 _H | 32 | FFC5B01C _H | 0 | 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM DataECC Control Register | IDCECCTL_PE1 | 00000000 _H | 32 | FFC60400 _H | 0 | 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM ErrorInformation Control Register | IDERRINT_PE1 | 00000003 _H | 32 | FFC60404 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM DataECC SED Status Clear Register | IDSERSTCLR_PE1 | 00000000 _H | 32 | FFC60408 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM DataECC DED Status Clear Register | IDDERSTCLR_PE1 | 00000000 _H | 32 | FFC6040C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM ErrorCount Overflow Status Register | IDOVFSTR_PE1 | 00000000 _H | 32 | FFC60410 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM DataECC SED Status Register (PE1) | IDSERSTR_PE1 | 00000000 _H | 32 | FFC60420 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM DataECC DED Status Register | IDDERSTR_PE1 | 00000000 _H | 32 | FFC60450 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM(Bank0) Data ECC SED Address Register | IDSEDADR0_PE1 | 00000000 _H | 32 | FFC60460 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM(Bank1) Data ECC SED Address Register | IDSEDADR1_PE1 | 00000000 _H | 32 | FFC60464 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM(Bank0) Data ECC DED Address Register | IDDEDADR0_PE1 | 00000000 _H | 32 | FFC604E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Data RAM(Bank1) Data ECC DED Address Register | IDDEDADR1_PE1 | 00000000 _H | 32 | FFC604E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM ECC Control Register | ITECCCTL_PE1 | 00000000 _H | 32 | FFC61400 _H | 0 | 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM Error Information Control Register | ITERRINT_PE1 | 00000003 _H | 32 | FFC61404 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM ECC SED Status Clear Register | ITSERSTCLR_PE1 | 00000000 _H | 32 | FFC61408 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM ECC DED Status Clear Register | ITDERSTCLR_PE1 | 00000000 _H | 32 | FFC6140C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM Error Count Overflow Status Register | ITOVFSTR_PE1 | XXXXXXXX _H | 32 | FFC61410 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM ECC SED Status Register (PE1) | ITSERSTR_PE1 | 00000000 _H | 32 | FFC61420 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM ECC DED Status Register | ITDERSTR_PE1 | 00000000 _H | 32 | FFC61450 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM SED Address Register | ITSEDADR_PE1 | 00000000 _H | 32 | FFC61460 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCIC1 | Instruction Cache Tag RAM DED Address Register | ITDEDADR_PE1 | 00000000 _H | 32 | FFC614E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash Error Information Control Register | UCFERRINT | 00000047 _H | 32 | FFC62000 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash ECC SED Status Clear Register | UCFSERSTCLR | 00000000 _H | 32 | FFC62004 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash ECC DED/Address Parity Error Status Clear Register | UCFDERSTCLR | 00000000 _H | 32 | FFC62008 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash Error Count Overflow Status Register | UCFOVFSTR | 00000000 _H | 32 | FFC6200C _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash ECC SED Status Register | UCFSERSTR | 00000000 _H | 32 | FFC62020 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash ECC DED/Address Parity Error Status Register | UCFDERSTR | 00000000 _H | 32 | FFC62030 _H | 0 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash 1 ECC SED Address Register | UCF1SEDADR | 00000000 _H | 32 | FFC62040 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLI | Code Flash 2 ECC SED Address Register | UCF2SEDADR | 00000000 _H | 32 | FFC62044 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (22/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|---------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ECCFLI | Code Flash 3 ECC SED Address Register | UCF3SEDADR | 00000000 _H | 32 | FFC62048 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCFLI | Code Flash 4 ECC SED Address Register | UCF4SEDADR | 00000000 _H | 32 | FFC6204C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCFLI | Code Flash ECC DED/Address Parity Error Address Register | UCFDEDADR | 00000000 _H | 32 | FFC620C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCFLI | Code Flash Address Parity Control Register | CFAPCTL | 00000000 _H | 32 | FFC62100 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCFLI | Code Flash ECC Control Register | CFECCCTL_VCI2CFBA | 00000000 _H | 32 | FFC62200 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCFLI | Code Flash Sub-Test Control Register | CFSTSTCTL_VCI2CFBA | 00000000 _H | 32 | FFC622F0 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCFLI | Code Flash ECC Control Register | CFECCCTL_PE1 | 00000000 _H | 32 | FFC62400 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCFLI | Code Flash Sub-Test Control Register | CFSTSTCTL_PE1 | 00000000 _H | 32 | FFC624F0 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | Code Flash Address ECC Control Register | CFECCCTL_VCI2CFBB | 00000000 _H | 32 | FFC62C00 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | Code Flash Address Error Information Control Register | CFERRINT_VCI2CFBB | 00000030 _H | 32 | FFC62C04 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | Code Flash Address ECC SED/DED Status Clear Register | CFERSTCLR_VCI2CFBB | 00000000 _H | 32 | FFC62C08 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | Code Flash Address Error Count Overflow Status Register | CFOVFSTR_VCI2CFBB | 00000000 _H | 32 | FFC62C0C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | Code Flash Address ECC SED/DED Status Register | CFERSTR_VCI2CFBB | 00000000 _H | 32 | FFC62C10 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | Code Flash Address ECC SED/DED Address Register | CFEADR0_VCI2CFBB | 00000000 _H | 32 | FFC62C50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | IFU Data ECC Control Register | IFECCCTL_PE1 | 00000000 _H | 32 | FFC63400 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | IFU Data Error Information Control Register | IFERRINT_PE1 | 00000003 _H | 32 | FFC63404 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | IFU Data ECC SED/DED Status Clear Register | IFERSTCLR_PE1 | 00000000 _H | 32 | FFC63408 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | IFU Data Error Count Overflow Status Register | IFOVFSTR_PE1 | 00000000 _H | 32 | FFC6340C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | IFU Data ECC SED/DED Status Register | IFERSTR_PE1 | 00000000 _H | 32 | FFC63410 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | IFU Data ECC SED/DED Address Register | IFERADR_PE1 | 00000000 _H | 32 | FFC63450 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM Error Information Control Register | UGRERRINT | 00000073 _H | 32 | FFC64000 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global-RAM ECC SED Status Clear Register | UGRSERSTCLR | 00000000 _H | 32 | FFC64004 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM ECC DED Status Clear Register | UGRDERSTCLR | 00000000 _H | 32 | FFC64008 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM Error Count Overflow Status Register | UGROVFSTR | 00000000 _H | 32 | FFC6400C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM ECC SED Status Register | UGRSERSTR | 00000000 _H | 32 | FFC64020 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM ECC DED Status Register | UGRDERSTR | 00000000 _H | 32 | FFC64030 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 1 ECC SED Address Register | UGR1SEDADR | 00000000 _H | 32 | FFC64040 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 2 ECC SED Address Register | UGR2SEDADR | 00000000 _H | 32 | FFC64044 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 3 ECC SED Address Register | UGR3SEDADR | 00000000 _H | 32 | FFC64048 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 4 ECC SED Address Register | UGR4SEDADR | 00000000 _H | 32 | FFC6404C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 5 ECC SED Address Register | UGR5SEDADR | 00000000 _H | 32 | FFC64050 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 6 ECC SED Address Register | UGR6SEDADR | 00000000 _H | 32 | FFC64054 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 7 ECC SED Address Register | UGR7SEDADR | 00000000 _H | 32 | FFC64058 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 8 ECC SED Address Register | UGR8SEDADR | 00000000 _H | 32 | FFC6405C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 9 ECC SED Address Register | UGR9SEDADR | 00000000 _H | 32 | FFC64060 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 10 ECC SED Address Register | UGR10SEDADR | 00000000 _H | 32 | FFC64064 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 11 ECC SED Address Register | UGR11SEDADR | 00000000 _H | 32 | FFC64068 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 12 ECC SED Address Register | UGR12SEDADR | 00000000 _H | 32 | FFC6406C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 13 ECC SED Address Register | UGR13SEDADR | 00000000 _H | 32 | FFC64070 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 14 ECC SED Address Register | UGR14SEDADR | 00000000 _H | 32 | FFC64074 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 15 ECC SED Address Register | UGR15SEDADR | 00000000 _H | 32 | FFC64078 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 16 ECC SED Address Register | UGR16SEDADR | 00000000 _H | 32 | FFC6407C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 17 ECC SED Address Register | UGR17SEDADR | 00000000 _H | 32 | FFC64080 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 18 ECC SED Address Register | UGR18SEDADR | 00000000 _H | 32 | FFC64084 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 19 ECC SED Address Register | UGR19SEDADR | 00000000 _H | 32 | FFC64088 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 20 ECC SED Address Register | UGR20SEDADR | 00000000 _H | 32 | FFC6408C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (23/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|---------|---|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ECCGRAM | Global RAM 21 ECC SED Address Register | UGR21SEDADR | 00000000 _H | 32 | FFC64090 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 22 ECC SED Address Register | UGR22SEDADR | 00000000 _H | 32 | FFC64094 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 23 ECC SED Address Register | UGR23SEDADR | 00000000 _H | 32 | FFC64098 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 24 ECC SED Address Register | UGR24SEDADR | 00000000 _H | 32 | FFC6409C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 25 ECC SED Address Register | UGR25SEDADR | 00000000 _H | 32 | FFC640A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 26 ECC SED Address Register | UGR26SEDADR | 00000000 _H | 32 | FFC640A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 27 ECC SED Address Register | UGR27SEDADR | 00000000 _H | 32 | FFC640A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 28 ECC SED Address Register | UGR28SEDADR | 00000000 _H | 32 | FFC640AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 29 ECC SED Address Register | UGR29SEDADR | 00000000 _H | 32 | FFC640B0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 30 ECC SED Address Register | UGR30SEDADR | 00000000 _H | 32 | FFC640B4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 31 ECC SED Address Register | UGR31SEDADR | 00000000 _H | 32 | FFC640B8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM 32 ECC SED Address Register | UGR32SEDADR | 00000000 _H | 32 | FFC640BC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM ECC DED Address Register | UGR00DEDADR | 00000000 _H | 32 | FFC640C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM ECC Control Register | GRECCCTL_GRAMC | 00000000 _H | 32 | FFC64100 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM Test Control Register | GRTSTCTL | 00000000 _H | 32 | FFC64104 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM Test Data Read Buffer (Lower 32 bit of Bank0) | GRDATBF0L | 00000000 _H | 32 | FFC64108 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM Test Data Read Buffer (Upper 32 bit of Bank0) | GRDATBF0H | 00000000 _H | 32 | FFC6410C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM Test Data Read Buffer (Lower 32 bit of Bank 1) | GRDATBF1L | 00000000 _H | 32 | FFC64110 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global-RAM Test Data Read Buffer (Upper 32 bit of Bank 1) | GRDATBF1H | 00000000 _H | 32 | FFC64114 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | GRAMC ECC Decoder Input Data Buffer 0 | GRDECINBF0 | 00000000 _H | 32 | FFC64118 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | GRAMC ECC Decoder Input Data Buffer 1 | GRDECINBF1 | 00000000 _H | 32 | FFC6411C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM ECC Control Register | GRECCCTL_VCI2GRAM | 00000000 _H | 32 | FFC64200 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCGRAM | Global RAM ECC Control Register | GRECCCTL_AXI2GRAM | 00000000 _H | 32 | FFC64E00 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM ECC Control Register | LRECCCTL_PE1 | 00000000 _H | 32 | FFC65400 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM error information control register | LRERRINT_PE1 | 00000043 _H | 32 | FFC65404 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM ECC SED Status Clear Register | LRSERSTCLR_PE1 | 00000000 _H | 32 | FFC65408 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM ECC SED Status Clear Register | LRDERSTCLR_PE1 | 00000000 _H | 32 | FFC6540C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM error count overflow status register | LROVFSTR_PE1 | 00000000 _H | 32 | FFC65410 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM ECC SED Status Register | LRSERSTR_PE1 | 00000000 _H | 32 | FFC65440 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM ECC SED Status Register | LRDERSTR_PE1 | 00000000 _H | 32 | FFC65450 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 1st error address register 0 | LR1SEDADR0_PE1 | 00000000 _H | 32 | FFC65460 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 1st error address register 1 | LR1SEDADR1_PE1 | 00000000 _H | 32 | FFC65464 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 1st error address register 2 | LR1SEDADR2_PE1 | 00000000 _H | 32 | FFC65468 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 1st error address register 3 | LR1SEDADR3_PE1 | 00000000 _H | 32 | FFC6546C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 2nd error address register 0 | LR2SEDADR0_PE1 | 00000000 _H | 32 | FFC65470 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 2nd error address register 1 | LR2SEDADR1_PE1 | 00000000 _H | 32 | FFC65474 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 2nd error address register 2 | LR2SEDADR2_PE1 | 00000000 _H | 32 | FFC65478 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 2nd error address register 3 | LR2SEDADR3_PE1 | 00000000 _H | 32 | FFC6547C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 3rd error address register 0 | LR3SEDADR0_PE1 | 00000000 _H | 32 | FFC65480 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 3rd error address register 1 | LR3SEDADR1_PE1 | 00000000 _H | 32 | FFC65484 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 3rd error address register 2 | LR3SEDADR2_PE1 | 00000000 _H | 32 | FFC65488 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 3rd error address register 3 | LR3SEDADR3_PE1 | 00000000 _H | 32 | FFC6548C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 4th error address register 0 | LR4SEDADR0_PE1 | 00000000 _H | 32 | FFC65490 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 4th error address register 1 | LR4SEDADR1_PE1 | 00000000 _H | 32 | FFC65494 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 4th error address register 2 | LR4SEDADR2_PE1 | 00000000 _H | 32 | FFC65498 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 4th error address register 3 | LR4SEDADR3_PE1 | 00000000 _H | 32 | FFC6549C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (24/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|---------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ECCCPU1 | Local RAM 5th error address register 0 | LR5SEDADR0_PE1 | 00000000 _H | 32 | FFC654A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 5th error address register 1 | LR5SEDADR1_PE1 | 00000000 _H | 32 | FFC654A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 5th error address register 2 | LR5SEDADR2_PE1 | 00000000 _H | 32 | FFC654A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 5th error address register 3 | LR5SEDADR3_PE1 | 00000000 _H | 32 | FFC654AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 6th error address register 0 | LR6SEDADR0_PE1 | 00000000 _H | 32 | FFC654B0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 6th error address register 1 | LR6SEDADR1_PE1 | 00000000 _H | 32 | FFC654B4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 6th error address register 2 | LR6SEDADR2_PE1 | 00000000 _H | 32 | FFC654B8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 6th error address register 3 | LR6SEDADR3_PE1 | 00000000 _H | 32 | FFC654BC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 7th error address register 0 | LR7SEDADR0_PE1 | 00000000 _H | 32 | FFC654C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 7th error address register 1 | LR7SEDADR1_PE1 | 00000000 _H | 32 | FFC654C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 7th error address register 2 | LR7SEDADR2_PE1 | 00000000 _H | 32 | FFC654C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 7th error address register 3 | LR7SEDADR3_PE1 | 00000000 _H | 32 | FFC654CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 8th error address register 0 | LR8SEDADR0_PE1 | 00000000 _H | 32 | FFC654D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 8th error address register 1 | LR8SEDADR1_PE1 | 00000000 _H | 32 | FFC654D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 8th error address register 2 | LR8SEDADR2_PE1 | 00000000 _H | 32 | FFC654D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local RAM 8th error address register 3 | LR8SEDADR3_PE1 | 00000000 _H | 32 | FFC654DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM DED Address Register 0 | LRDEDADR0_PE1 | 00000000 _H | 32 | FFC654E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM DED Address Register 1 | LRDEDADR1_PE1 | 00000000 _H | 32 | FFC654E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM DED Address Register 2 | LRDEDADR2_PE1 | 00000000 _H | 32 | FFC654E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM DED Address Register 3 | LRDEDADR3_PE1 | 00000000 _H | 32 | FFC654EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM Test Control Register | LRTSTCTL_PE1 | 00000000 _H | 32 | FFC65604 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM Test Data Read Buffer 0 | LRTDATBF0_PE1 | 00000000 _H | 32 | FFC65608 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCPU1 | Local-RAM Test Data Read Buffer 1 | LRTDATBF1_PE1 | 00000000 _H | 32 | FFC6560C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Slave Data and Address ECC Control Register | LSSECCCTL_PE1 | 00000000 _H | 32 | FFC66000 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Slave Data and Address Error Information Control Register | LSSERRINT_PE1 | 00000033 _H | 32 | FFC66004 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Slave Data and Address ECC SED/DED Status Clear Register | LSSERSTCLR_PE1 | 00000000 _H | 32 | FFC66008 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Slave Data and Address Error Count Overflow Status Register | LSSOVFSTR_PE1 | 00000000 _H | 32 | FFC6600C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Slave Data and Address ECC SED/DED Status Register | LSSERSTR_PE1 | 00000000 _H | 32 | FFC66010 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Slave Data and Address ECC SED/DED Address Register | LSSEADR_PE1 | 00000000 _H | 32 | FFC66050 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Master Data ECC Control Register | LSMECCCTL_PE1 | 00000000 _H | 32 | FFC66400 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Master Data Error Information Control Register | LSMERRINT_PE1 | 00000003 _H | 32 | FFC66404 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Master Data ECC SED/DED Status Clear Register | LSMERSTCLR_PE1 | 00000000 _H | 32 | FFC66408 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Master Data Error Count Overflow Status Register | LSMOVFSTR_PE1 | 00000000 _H | 32 | FFC6640C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Master Data ECC SED/DED Status Register | LSMERSTR_PE1 | 00000000 _H | 32 | FFC66410 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | LSU Master Data ECC SED/DED Address Register | LSMEADR_PE1 | 00000000 _H | 32 | FFC66450 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data Error Information Control Register | VPERRINT_SG0 | 00000003 _H | 32 | FFC67004 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Clear Register | VPERSTCLR_SG0 | 00000000 _H | 32 | FFC67008 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data Error Count Overflow Status Register | VPOVFSTR_SG0 | 00000000 _H | 32 | FFC6700C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Register | VPERSTR_SG0 | 00000000 _H | 32 | FFC67010 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Address Register | VPEADR_SG0 | 00000000 _H | 32 | FFC67050 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data Error Information Control Register | VPERRINT_SG1 | 00000003 _H | 32 | FFC67404 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Clear Register | VPERSTCLR_SG1 | 00000000 _H | 32 | FFC67408 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (25/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|----------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBECC0 | System interconnect (PBus I/F) Data Error Count Overflow Status Register | VPOVFSTR_SG1 | 00000000 _H | 32 | FFC6740C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Register | VPERSTR_SG1 | 00000000 _H | 32 | FFC67410 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Address Register | VPEADR_SG1 | 00000000 _H | 32 | FFC67450 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data Error Information Control Register | VPERRINT_SG2 | 00000003 _H | 32 | FFC67804 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Clear Register | VPERSTCLR_SG2 | 00000000 _H | 32 | FFC67808 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data Error Count Overflow Status Register | VPOVFSTR_SG2 | 00000000 _H | 32 | FFC6780C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Register | VPERSTR_SG2 | 00000000 _H | 32 | FFC67810 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Address Register | VPEADR_SG2 | 00000000 _H | 32 | FFC67850 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data Error Information Control Register | VPERRINT_SG5 | 00000003 _H | 32 | FFC68404 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Clear Register | VPERSTCLR_SG5 | 00000000 _H | 32 | FFC68408 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data Error Count Overflow Status Register | VPOVFSTR_SG5 | 00000000 _H | 32 | FFC6840C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Status Register | VPERSTR_SG5 | 00000000 _H | 32 | FFC68410 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System interconnect (PBus I/F) Data ECC SED/DED Address Register | VPEADR_SG5 | 00000000 _H | 32 | FFC68450 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System Interconnect Data ECC Control Register | VCECCCTL_PDMA | 00000000 _H | 32 | FFC6A000 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System Interconnect Error Information Control Register | VCERRINT_PDMA | 00000003 _H | 32 | FFC6A004 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System Interconnect Data ECC SED/DED Status Clear Register | VCERSTCLR_PDMA | 00000000 _H | 32 | FFC6A008 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System Interconnect Data Error Count Overflow Status Register | VCOVFSTR_PDMA | 00000000 _H | 32 | FFC6A00C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System Interconnect Data ECC SED/DED Status Register | VCERSTR_PDMA | 00000000 _H | 32 | FFC6A010 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | System Interconnect Data ECC SED/DED Address Register | VCEADR_PDMA | 00000000 _H | 32 | FFC6A050 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | P-Bus Data and Address ECC Control Register | APECCCTL_PFSS | 00000000 _H | 32 | FFC6C000 _H | 0 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | P-Bus Error Information Control Register | APERRINT_PFSS | 00000073 _H | 32 | FFC6C004 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | P-Bus Data and Address ECC SED/DED Status Clear Register | APERSTCLR_PFSS | 00000000 _H | 32 | FFC6C008 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | P-Bus Data and Address Error Count Overflow Status Register | APOVFSTR_PFSS | 00000000 _H | 32 | FFC6C00C _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | P-Bus Data and Address ECC SED/DED Status Register | APERSTR_PFSS | 00000000 _H | 32 | FFC6C010 _H | 0 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC0 | P-Bus Data and Address ECC SED/DED Address Register | APEADR_PFSS | 00000000 _H | 32 | FFC6C050 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH0 | ECC control register | ECCCSIH0CTL | 00000X1X _H | 32 | FFC70000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH0 | ECC test mode control register | ECCCSIH0TMC | 00000000 _H | 32 | FFC70004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH0 | ECC bit data control test register | ECCCSIH0TRC | 00000000 _H | 32 | FFC70008 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH0 | ECC encoder and decoder data test register | ECCCSIH0TED | 00000000 _H | 32 | FFC7000C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH0 | ECC error address register | ECCCSIH0EAD0 | 00000000 _H | 32 | FFC70010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH1 | ECC control register | ECCCSIH1CTL | 00000X1X _H | 32 | FFC70040 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH1 | ECC test mode control register | ECCCSIH1TMC | 00000000 _H | 32 | FFC70044 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH1 | ECC bit data control test register | ECCCSIH1TRC | 00000000 _H | 32 | FFC70048 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH1 | ECC encoder and decoder data test register | ECCCSIH1TED | 00000000 _H | 32 | FFC7004C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH1 | ECC error address register | ECCCSIH1EAD0 | 00000000 _H | 32 | FFC70050 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH2 | ECC control register | ECCCSIH2CTL | 00000X1X _H | 32 | FFC70080 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH2 | ECC test mode control register | ECCCSIH2TMC | 00000000 _H | 32 | FFC70084 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH2 | ECC bit data control test register | ECCCSIH2TRC | 00000000 _H | 32 | FFC70088 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH2 | ECC encoder and decoder data test register | ECCCSIH2TED | 00000000 _H | 32 | FFC7008C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECCCSIH2 | ECC error address register | ECCCSIH2EAD0 | 00000000 _H | 32 | FFC70090 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (26/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|-----------|---|---------------|---|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ECCSIH3 | ECC control register | ECCSIH3CTL | 0000X1X _H | 32 | FFC700C0 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCSIH3 | ECC test mode control register | ECCSIH3TMC | 00000000 _H | 32 | FFC700C4 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCSIH3 | ECC bit data control test register | ECCSIH3TRC | 00000000 _H | 32 | FFC700C8 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCSIH3 | ECC encoder and decoder data test register | ECCSIH3TED | 00000000 _H | 32 | FFC700CC _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCSIH3 | ECC error address register | ECCSIH3EAD0 | 00000000 _H | 32 | FFC700D0 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN0 | ECC control register | ECCCAN0CTL | 0000X1X _H | 32 | FFC71100 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN0 | ECC test mode control register | ECCCAN0TMC | 00000000 _H | 32 | FFC71104 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN0 | ECC bit data control test register | ECCCAN0TRC | 00000000 _H | 32 | FFC71108 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN0 | ECC encoder and decoder data test register | ECCCAN0TED | 00000000 _H | 32 | FFC7110C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN0 | ECC error address register | ECCCAN0EAD0 | 00000000 _H | 32 | FFC71110 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN1 | ECC control register | ECCCAN1CTL | 0000X1X _H | 32 | FFC71200 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN1 | ECC test mode control register | ECCCAN1TMC | 00000000 _H | 32 | FFC71204 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN1 | ECC bit data control test register | ECCCAN1TRC | 00000000 _H | 32 | FFC71208 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN1 | ECC encoder and decoder data test register | ECCCAN1TED | 00000000 _H | 32 | FFC7120C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCCAN1 | ECC error address register | ECCCAN1EAD0 | 00000000 _H | 32 | FFC71210 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0 | ECC control register | ECCFLX0CTL | 0000X1X _H | 32 | FFC72000 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0 | ECC test mode control register | ECCFLX0TMC | 00000000 _H | 32 | FFC72004 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0 | ECC bit data control test register | ECCFLX0TRC | 00000000 _H | 32 | FFC72008 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0 | ECC encoder and decoder data test register | ECCFLX0TED | 00000000 _H | 32 | FFC7200C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0 | ECC error address register | ECCFLX0EAD0 | 00000000 _H | 32 | FFC72010 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T1 | ECC control register | ECCFLX0T1CTL | 0000X1X _H | 32 | FFC72040 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T1 | ECC test mode control register | ECCFLX0T1TMC | 00000000 _H | 32 | FFC72044 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T1 | ECC bit data control test register | ECCFLX0T1TRC | 00000000 _H | 32 | FFC72048 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T1 | ECC encoder and decoder data test register | ECCFLX0T1TED | 00000000 _H | 32 | FFC7204C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T1 | ECC error address register | ECCFLX0T1EAD0 | 00000000 _H | 32 | FFC72050 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T0 | ECC control register | ECCFLX0T0CTL | 0000X1X _H | 32 | FFC72080 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T0 | ECC test mode control register | ECCFLX0T0TMC | 00000000 _H | 32 | FFC72084 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T0 | ECC bit data control test register | ECCFLX0T0TRC | 00000000 _H | 32 | FFC72088 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T0 | ECC encoder and decoder data test register | ECCFLX0T0TED | 00000000 _H | 32 | FFC7208C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECCFLX0T0 | ECC error address register | ECCFLX0T0EAD0 | 00000000 _H | 32 | FFC72090 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| FLASH | Reset Vector 0 Register | GREG8 | Specified by the user | 32 | FFCD0020 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| FLASH | Option Byte 0 | OPBT0 | Specified by the user | 32 | FFCD0030 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| FLASH | Option Byte 2 | OPBT2 | Specified by the user | 32 | FFCD0038 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| FLASH | Product name storage register | PRDNAME1 | XXXXXXXX _H | 32 | FFCD00D0 _H | 5 | 32 | — | — | — | — | √ | √ |
| FLASH | Product name storage register | PRDNAME2 | XXXXXXXX _H | 32 | FFCD00D4 _H | 5 | 32 | — | — | — | — | √ | √ |
| FLASH | Product name storage register | PRDNAME3 | XXXXXXXX _H | 32 | FFCD00D8 _H | 5 | 32 | — | — | — | — | √ | √ |
| FLASH | Product name storage register | PRDNAME4 | XXXXXXXX _H | 32 | FFCD00DC _H | 5 | 32 | — | — | — | — | √ | √ |
| FLASH | Temperature sensor reference temperature storage register | TSNREFD | Undefined: (The fixed value is set at the time of shipping.) | 32 | FFCD019C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Logic BIST reference value register 1 | LBISTREF1 | 00A5A5A _H | 32 | FFCDA000 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | Logic BIST reference value register 2 | LBISTREF2 | 0005A5A5 _H | 32 | FFCDA004 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | Memory BIST reference value register | MBISTREF | 00A5A5A _H | 32 | FFCDA008 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | Logic BIST signature value register 1 | LBISTSIG1 | 0005A5A5 _H | 32 | FFCDA010 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | Logic BIST signature value register 2 | LBISTSIG2 | 00A5A5A _H | 32 | FFCDA014 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | Memory BIST signature value register | MBISTSIG | 00055AA5 _H | 32 | FFCDA018 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | Memory BIST FTAG signature value register L | MBISTFTAGL | FFFFFFFF _H | 32 | FFCDA020 _H | 5 | 32 | √ | √ | √ | — | √ | √ |

Table A.1 List of Registers (27/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| SYS | Memory BIST FTAG signature value register H | MBISTFTAGH | FFFFFFFH _H | 32 | FFCDA024 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| SYS | BIST Sequence status register | BSEQ0ST | 00000001 _H | 32 | FFCDA030 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| SYS | BIST Sequence inverted status register | BSEQ0STB | 00000002 _H | 32 | FFCDA034 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| SYS | Field BIST Result register | BISTST | 00000007 _H | 32 | FFCDA038 _H | 5 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| RSCAN0 | Channel 0 configuration register | RSCAN0C0CFG | 00000000 _H | 32 | FFD20000 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 0 control register | RSCAN0C0CTR | 00000005 _H | 32 | FFD20004 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 0 status register | RSCAN0C0STS | 00000005 _H | 32 | FFD20008 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 0 error flag register | RSCAN0C0ERFL | 00000000 _H | 32 | FFD2000C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 1 configuration register | RSCAN0C1CFG | 00000000 _H | 32 | FFD20010 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 1 control register | RSCAN0C1CTR | 00000005 _H | 32 | FFD20014 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 1 status register | RSCAN0C1STS | 00000005 _H | 32 | FFD20018 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 1 error flag register | RSCAN0C1ERFL | 00000000 _H | 32 | FFD2001C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 2 configuration register | RSCAN0C2CFG | 00000000 _H | 32 | FFD20020 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 2 control register | RSCAN0C2CTR | 00000005 _H | 32 | FFD20024 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 2 status register | RSCAN0C2STS | 00000005 _H | 32 | FFD20028 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Channel 2 error flag register | RSCAN0C2ERFL | 00000000 _H | 32 | FFD2002C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global configuration register | RSCAN0GCFG | 00000000 _H | 32 | FFD20084 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global control register | RSCAN0GCTR | 00000005 _H | 32 | FFD20088 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global status register | RSCAN0GSTS | 0000000D _H | 32 | FFD2008C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global error flag register | RSCAN0GERFL | 00000000 _H | 32 | FFD20090 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global timestamp counter register | RSCAN0GTSC | 00000000 _H | 32 | FFD20094 _H | 2 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule entry control register | RSCAN0GAFLECTR | 00000000 _H | 32 | FFD20098 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule configuration register 0 | RSCAN0GALCFG0 | 00000000 _H | 32 | FFD2009C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer number register | RSCAN0RMNB | 00000000 _H | 32 | FFD200A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer new data register 0 | RSCAN0RMND0 | 00000000 _H | 32 | FFD200A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer new data register 1 | RSCAN0RMND1 | 00000000 _H | 32 | FFD200AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 0 | RSCAN0RFCC0 | 00000000 _H | 32 | FFD200B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 1 | RSCAN0RFCC1 | 00000000 _H | 32 | FFD200BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 2 | RSCAN0RFCC2 | 00000000 _H | 32 | FFD200C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 3 | RSCAN0RFCC3 | 00000000 _H | 32 | FFD200C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 4 | RSCAN0RFCC4 | 00000000 _H | 32 | FFD200C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 5 | RSCAN0RFCC5 | 00000000 _H | 32 | FFD200CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 6 | RSCAN0RFCC6 | 00000000 _H | 32 | FFD200D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer configuration and control register 7 | RSCAN0RFCC7 | 00000000 _H | 32 | FFD200D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 0 | RSCAN0RFSTS0 | 00000001 _H | 32 | FFD200D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 1 | RSCAN0RFSTS1 | 00000001 _H | 32 | FFD200DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 2 | RSCAN0RFSTS2 | 00000001 _H | 32 | FFD200E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 3 | RSCAN0RFSTS3 | 00000001 _H | 32 | FFD200E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 4 | RSCAN0RFSTS4 | 00000001 _H | 32 | FFD200E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 5 | RSCAN0RFSTS5 | 00000001 _H | 32 | FFD200EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 6 | RSCAN0RFSTS6 | 00000001 _H | 32 | FFD200F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer status register 7 | RSCAN0RFSTS7 | 00000001 _H | 32 | FFD200F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer pointer control register 0 | RSCAN0RFPCTR0 | 00000000 _H | 32 | FFD200F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer pointer control register 1 | RSCAN0RFPCTR1 | 00000000 _H | 32 | FFD200FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer pointer control register 2 | RSCAN0RFPCTR2 | 00000000 _H | 32 | FFD20100 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (28/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Receive FIFO buffer pointer control register 3 | RSCAN0RFPCTR3 | 00000000 _H | 32 | FFD20104 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive FIFO buffer pointer control register 4 | RSCAN0RFPCTR4 | 00000000 _H | 32 | FFD20108 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive FIFO buffer pointer control register 5 | RSCAN0RFPCTR5 | 00000000 _H | 32 | FFD2010C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive FIFO buffer pointer control register 6 | RSCAN0RFPCTR6 | 00000000 _H | 32 | FFD20110 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive FIFO buffer pointer control register 7 | RSCAN0RFPCTR7 | 00000000 _H | 32 | FFD20114 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 0 | RSCAN0CFCC0 | 00000000 _H | 32 | FFD20118 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 1 | RSCAN0CFCC1 | 00000000 _H | 32 | FFD2011C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 2 | RSCAN0CFCC2 | 00000000 _H | 32 | FFD20120 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 3 | RSCAN0CFCC3 | 00000000 _H | 32 | FFD20124 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 4 | RSCAN0CFCC4 | 00000000 _H | 32 | FFD20128 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 5 | RSCAN0CFCC5 | 00000000 _H | 32 | FFD2012C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 6 | RSCAN0CFCC6 | 00000000 _H | 32 | FFD20130 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 7 | RSCAN0CFCC7 | 00000000 _H | 32 | FFD20134 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer configuration and control register 8 | RSCAN0CFCC8 | 00000000 _H | 32 | FFD20138 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 0 | RSCAN0CFSTS0 | 00000001 _H | 32 | FFD20178 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 1 | RSCAN0CFSTS1 | 00000001 _H | 32 | FFD2017C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 2 | RSCAN0CFSTS2 | 00000001 _H | 32 | FFD20180 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 3 | RSCAN0CFSTS3 | 00000001 _H | 32 | FFD20184 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 4 | RSCAN0CFSTS4 | 00000001 _H | 32 | FFD20188 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 5 | RSCAN0CFSTS5 | 00000001 _H | 32 | FFD2018C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 6 | RSCAN0CFSTS6 | 00000001 _H | 32 | FFD20190 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 7 | RSCAN0CFSTS7 | 00000001 _H | 32 | FFD20194 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer status register 8 | RSCAN0CFSTS8 | 00000001 _H | 32 | FFD20198 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 0 | RSCAN0CFPCTR0 | 00000000 _H | 32 | FFD201D8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 1 | RSCAN0CFPCTR1 | 00000000 _H | 32 | FFD201DC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 2 | RSCAN0CFPCTR2 | 00000000 _H | 32 | FFD201E0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 3 | RSCAN0CFPCTR3 | 00000000 _H | 32 | FFD201E4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 4 | RSCAN0CFPCTR4 | 00000000 _H | 32 | FFD201E8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 5 | RSCAN0CFPCTR5 | 00000000 _H | 32 | FFD201EC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 6 | RSCAN0CFPCTR6 | 00000000 _H | 32 | FFD201F0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 7 | RSCAN0CFPCTR7 | 00000000 _H | 32 | FFD201F4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer pointer control register 8 | RSCAN0CFPCTR8 | 00000000 _H | 32 | FFD201F8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | FIFO empty status register | RSCAN0FESTS | 03FFFFFF _H | 32 | FFD20238 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | FIFO full status register | RSCAN0FFSTS | 00000000 _H | 32 | FFD2023C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | FIFO message lost status register | RSCAN0FMSTS | 00000000 _H | 32 | FFD20240 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive FIFO buffer interrupt flag status register | RSCAN0RFISTS | 00000000 _H | 32 | FFD20244 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer receive interrupt flag status register | RSCAN0CFRISTS | 00000000 _H | 32 | FFD20248 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit/receive FIFO buffer transmit interrupt flag status register | RSCAN0CFTISTS | 00000000 _H | 32 | FFD2024C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 0 | RSCAN0TMC0 | 00 _H | 8 | FFD20250 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 1 | RSCAN0TMC1 | 00 _H | 8 | FFD20254 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 2 | RSCAN0TMC2 | 00 _H | 8 | FFD20258 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (29/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-------------------------------------|--------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit buffer control register 3 | RSCAN0TMC3 | 00 _H | 8 | FFD20253 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 4 | RSCAN0TMC4 | 00 _H | 8 | FFD20254 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 5 | RSCAN0TMC5 | 00 _H | 8 | FFD20255 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 6 | RSCAN0TMC6 | 00 _H | 8 | FFD20256 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 7 | RSCAN0TMC7 | 00 _H | 8 | FFD20257 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 8 | RSCAN0TMC8 | 00 _H | 8 | FFD20258 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 9 | RSCAN0TMC9 | 00 _H | 8 | FFD20259 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 10 | RSCAN0TMC10 | 00 _H | 8 | FFD2025A _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 11 | RSCAN0TMC11 | 00 _H | 8 | FFD2025B _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 12 | RSCAN0TMC12 | 00 _H | 8 | FFD2025C _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 13 | RSCAN0TMC13 | 00 _H | 8 | FFD2025D _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 14 | RSCAN0TMC14 | 00 _H | 8 | FFD2025E _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 15 | RSCAN0TMC15 | 00 _H | 8 | FFD2025F _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 16 | RSCAN0TMC16 | 00 _H | 8 | FFD20260 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 17 | RSCAN0TMC17 | 00 _H | 8 | FFD20261 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 18 | RSCAN0TMC18 | 00 _H | 8 | FFD20262 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 19 | RSCAN0TMC19 | 00 _H | 8 | FFD20263 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 20 | RSCAN0TMC20 | 00 _H | 8 | FFD20264 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 21 | RSCAN0TMC21 | 00 _H | 8 | FFD20265 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 22 | RSCAN0TMC22 | 00 _H | 8 | FFD20266 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 23 | RSCAN0TMC23 | 00 _H | 8 | FFD20267 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 24 | RSCAN0TMC24 | 00 _H | 8 | FFD20268 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 25 | RSCAN0TMC25 | 00 _H | 8 | FFD20269 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 26 | RSCAN0TMC26 | 00 _H | 8 | FFD2026A _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 27 | RSCAN0TMC27 | 00 _H | 8 | FFD2026B _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 28 | RSCAN0TMC28 | 00 _H | 8 | FFD2026C _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 29 | RSCAN0TMC29 | 00 _H | 8 | FFD2026D _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 30 | RSCAN0TMC30 | 00 _H | 8 | FFD2026E _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 31 | RSCAN0TMC31 | 00 _H | 8 | FFD2026F _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 32 | RSCAN0TMC32 | 00 _H | 8 | FFD20270 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 33 | RSCAN0TMC33 | 00 _H | 8 | FFD20271 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 34 | RSCAN0TMC34 | 00 _H | 8 | FFD20272 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 35 | RSCAN0TMC35 | 00 _H | 8 | FFD20273 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 36 | RSCAN0TMC36 | 00 _H | 8 | FFD20274 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 37 | RSCAN0TMC37 | 00 _H | 8 | FFD20275 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 38 | RSCAN0TMC38 | 00 _H | 8 | FFD20276 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 39 | RSCAN0TMC39 | 00 _H | 8 | FFD20277 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 40 | RSCAN0TMC40 | 00 _H | 8 | FFD20278 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 41 | RSCAN0TMC41 | 00 _H | 8 | FFD20279 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 42 | RSCAN0TMC42 | 00 _H | 8 | FFD2027A _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 43 | RSCAN0TMC43 | 00 _H | 8 | FFD2027B _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 44 | RSCAN0TMC44 | 00 _H | 8 | FFD2027C _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 45 | RSCAN0TMC45 | 00 _H | 8 | FFD2027D _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 46 | RSCAN0TMC46 | 00 _H | 8 | FFD2027E _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer control register 47 | RSCAN0TMC47 | 00 _H | 8 | FFD2027F _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 0 | RSCAN0TMSTS0 | 00 _H | 8 | FFD202D0 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 1 | RSCAN0TMSTS1 | 00 _H | 8 | FFD202D1 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 2 | RSCAN0TMSTS2 | 00 _H | 8 | FFD202D2 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (30/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit buffer status register 3 | RSCAN0TMSTS3 | 00 _H | 8 | FFD202D3 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 4 | RSCAN0TMSTS4 | 00 _H | 8 | FFD202D4 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 5 | RSCAN0TMSTS5 | 00 _H | 8 | FFD202D5 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 6 | RSCAN0TMSTS6 | 00 _H | 8 | FFD202D6 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 7 | RSCAN0TMSTS7 | 00 _H | 8 | FFD202D7 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 8 | RSCAN0TMSTS8 | 00 _H | 8 | FFD202D8 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 9 | RSCAN0TMSTS9 | 00 _H | 8 | FFD202D9 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 10 | RSCAN0TMSTS10 | 00 _H | 8 | FFD202DA _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 11 | RSCAN0TMSTS11 | 00 _H | 8 | FFD202DB _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 12 | RSCAN0TMSTS12 | 00 _H | 8 | FFD202DC _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 13 | RSCAN0TMSTS13 | 00 _H | 8 | FFD202DD _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 14 | RSCAN0TMSTS14 | 00 _H | 8 | FFD202DE _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 15 | RSCAN0TMSTS15 | 00 _H | 8 | FFD202DF _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 16 | RSCAN0TMSTS16 | 00 _H | 8 | FFD202E0 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 17 | RSCAN0TMSTS17 | 00 _H | 8 | FFD202E1 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 18 | RSCAN0TMSTS18 | 00 _H | 8 | FFD202E2 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 19 | RSCAN0TMSTS19 | 00 _H | 8 | FFD202E3 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 20 | RSCAN0TMSTS20 | 00 _H | 8 | FFD202E4 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 21 | RSCAN0TMSTS21 | 00 _H | 8 | FFD202E5 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 22 | RSCAN0TMSTS22 | 00 _H | 8 | FFD202E6 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 23 | RSCAN0TMSTS23 | 00 _H | 8 | FFD202E7 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 24 | RSCAN0TMSTS24 | 00 _H | 8 | FFD202E8 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 25 | RSCAN0TMSTS25 | 00 _H | 8 | FFD202E9 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 26 | RSCAN0TMSTS26 | 00 _H | 8 | FFD202EA _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 27 | RSCAN0TMSTS27 | 00 _H | 8 | FFD202EB _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 28 | RSCAN0TMSTS28 | 00 _H | 8 | FFD202EC _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 29 | RSCAN0TMSTS29 | 00 _H | 8 | FFD202ED _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 30 | RSCAN0TMSTS30 | 00 _H | 8 | FFD202EE _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 31 | RSCAN0TMSTS31 | 00 _H | 8 | FFD202EF _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 32 | RSCAN0TMSTS32 | 00 _H | 8 | FFD202F0 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 33 | RSCAN0TMSTS33 | 00 _H | 8 | FFD202F1 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 34 | RSCAN0TMSTS34 | 00 _H | 8 | FFD202F2 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 35 | RSCAN0TMSTS35 | 00 _H | 8 | FFD202F3 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 36 | RSCAN0TMSTS36 | 00 _H | 8 | FFD202F4 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 37 | RSCAN0TMSTS37 | 00 _H | 8 | FFD202F5 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 38 | RSCAN0TMSTS38 | 00 _H | 8 | FFD202F6 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 39 | RSCAN0TMSTS39 | 00 _H | 8 | FFD202F7 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 40 | RSCAN0TMSTS40 | 00 _H | 8 | FFD202F8 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 41 | RSCAN0TMSTS41 | 00 _H | 8 | FFD202F9 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 42 | RSCAN0TMSTS42 | 00 _H | 8 | FFD202FA _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 43 | RSCAN0TMSTS43 | 00 _H | 8 | FFD202FB _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 44 | RSCAN0TMSTS44 | 00 _H | 8 | FFD202FC _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 45 | RSCAN0TMSTS45 | 00 _H | 8 | FFD202FD _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 46 | RSCAN0TMSTS46 | 00 _H | 8 | FFD202FE _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer status register 47 | RSCAN0TMSTS47 | 00 _H | 8 | FFD202FF _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer transmit request status register 0 | RSCAN0TMTRSTS0 | 00000000 _H | 32 | FFD20350 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Transmit buffer transmit request status register 1 | RSCAN0TMTRSTS1 | 00000000 _H | 32 | FFD20354 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (31/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit buffer transmit abort request status register 0 | RSCAN0TMTARSTS0 | 00000000 _H | 32 | FFD20360 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer transmit abort request status register 1 | RSCAN0TMTARSTS1 | 00000000 _H | 32 | FFD20364 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer transmit complete status register 0 | RSCAN0TMTCASTS0 | 00000000 _H | 32 | FFD20370 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer transmit complete status register 1 | RSCAN0TMTCASTS1 | 00000000 _H | 32 | FFD20374 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer transmit abort status register 0 | RSCAN0TMTASTS0 | 00000000 _H | 32 | FFD20380 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer transmit abort status register 1 | RSCAN0TMTASTS1 | 00000000 _H | 32 | FFD20384 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer interrupt enable configuration register 0 | RSCAN0TMIEC0 | 00000000 _H | 32 | FFD20390 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer interrupt enable configuration register 1 | RSCAN0TMIEC1 | 00000000 _H | 32 | FFD20394 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue configuration and control register 0 | RSCAN0TXQCC0 | 00000000 _H | 32 | FFD203A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue configuration and control register 1 | RSCAN0TXQCC1 | 00000000 _H | 32 | FFD203A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue configuration and control register 2 | RSCAN0TXQCC2 | 00000000 _H | 32 | FFD203A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue status register 0 | RSCAN0TXQSTS0 | 00000001 _H | 32 | FFD203C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue status register 1 | RSCAN0TXQSTS1 | 00000001 _H | 32 | FFD203C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue status register 2 | RSCAN0TXQSTS2 | 00000001 _H | 32 | FFD203C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue pointer control register 0 | RSCAN0TXQPCTR0 | 00000000 _H | 32 | FFD203E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue pointer control register 1 | RSCAN0TXQPCTR1 | 00000000 _H | 32 | FFD203E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit queue pointer control register 2 | RSCAN0TXQPCTR2 | 00000000 _H | 32 | FFD203E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history configuration and control register 0 | RSCAN0THLCC0 | 00000000 _H | 32 | FFD20400 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history configuration and control register 1 | RSCAN0THLCC1 | 00000000 _H | 32 | FFD20404 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history configuration and control register 2 | RSCAN0THLCC2 | 00000000 _H | 32 | FFD20408 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history status register 0 | RSCAN0THLSTS0 | 00000001 _H | 32 | FFD20420 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history status register 1 | RSCAN0THLSTS1 | 00000001 _H | 32 | FFD20424 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history status register 2 | RSCAN0THLSTS2 | 00000001 _H | 32 | FFD20428 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history pointer control register 0 | RSCAN0THLPCTR0 | 00000000 _H | 32 | FFD20440 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history pointer control register 1 | RSCAN0THLPCTR1 | 00000000 _H | 32 | FFD20444 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history pointer control register 2 | RSCAN0THLPCTR2 | 00000000 _H | 32 | FFD20448 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global TX interrupt status register 0 | RSCAN0GTINTSTS0 | 00000000 _H | 32 | FFD20460 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global test configuration register | RSCAN0GTSTCFG | 00000000 _H | 32 | FFD20468 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global test control register | RSCAN0GTSTCTR | 00000000 _H | 32 | FFD2046C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global FD Configuration register | RSCAN0GFDCFG | 00000000 _H | 32 | FFD20474 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global lock key register | RSCAN0GLOCKK | 00000000 _H | 32 | FFD2047C _H | 2 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Global Register Map Configuration Register | RSCAN0GRMCFG | 00000000 _H | 32 | FFD204FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 0 | RSCAN0GAFLID0 | 00000000 _H | 32 | FFD20500 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 0 | RSCAN0GAFLM0 | 00000000 _H | 32 | FFD20504 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 0 | RSCAN0GAFLP0_0 | 00000000 _H | 32 | FFD20508 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 0 | RSCAN0GAFLP1_0 | 00000000 _H | 32 | FFD2050C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 1 | RSCAN0GAFLID1 | 00000000 _H | 32 | FFD20510 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 1 | RSCAN0GAFLM1 | 00000000 _H | 32 | FFD20514 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 1 | RSCAN0GAFLP0_1 | 00000000 _H | 32 | FFD20518 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 1 | RSCAN0GAFLP1_1 | 00000000 _H | 32 | FFD2051C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 2 | RSCAN0GAFLID2 | 00000000 _H | 32 | FFD20520 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 2 | RSCAN0GAFLM2 | 00000000 _H | 32 | FFD20524 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 2 | RSCAN0GAFLP0_2 | 00000000 _H | 32 | FFD20528 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 2 | RSCAN0GAFLP1_2 | 00000000 _H | 32 | FFD2052C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (32/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|------------------------------------|-----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Receive rule ID register 3 | RSCAN0GAFLID3 | 00000000 _H | 32 | FFD20530 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 3 | RSCAN0GAFLM3 | 00000000 _H | 32 | FFD20534 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 3 | RSCAN0GAFLP0_3 | 00000000 _H | 32 | FFD20538 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 3 | RSCAN0GAFLP1_3 | 00000000 _H | 32 | FFD2053C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 4 | RSCAN0GAFLID4 | 00000000 _H | 32 | FFD20540 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 4 | RSCAN0GAFLM4 | 00000000 _H | 32 | FFD20544 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 4 | RSCAN0GAFLP0_4 | 00000000 _H | 32 | FFD20548 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 4 | RSCAN0GAFLP1_4 | 00000000 _H | 32 | FFD2054C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 5 | RSCAN0GAFLID5 | 00000000 _H | 32 | FFD20550 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 5 | RSCAN0GAFLM5 | 00000000 _H | 32 | FFD20554 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 5 | RSCAN0GAFLP0_5 | 00000000 _H | 32 | FFD20558 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 5 | RSCAN0GAFLP1_5 | 00000000 _H | 32 | FFD2055C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 6 | RSCAN0GAFLID6 | 00000000 _H | 32 | FFD20560 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 6 | RSCAN0GAFLM6 | 00000000 _H | 32 | FFD20564 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 6 | RSCAN0GAFLP0_6 | 00000000 _H | 32 | FFD20568 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 6 | RSCAN0GAFLP1_6 | 00000000 _H | 32 | FFD2056C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 7 | RSCAN0GAFLID7 | 00000000 _H | 32 | FFD20570 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 7 | RSCAN0GAFLM7 | 00000000 _H | 32 | FFD20574 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 7 | RSCAN0GAFLP0_7 | 00000000 _H | 32 | FFD20578 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 7 | RSCAN0GAFLP1_7 | 00000000 _H | 32 | FFD2057C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 8 | RSCAN0GAFLID8 | 00000000 _H | 32 | FFD20580 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 8 | RSCAN0GAFLM8 | 00000000 _H | 32 | FFD20584 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 8 | RSCAN0GAFLP0_8 | 00000000 _H | 32 | FFD20588 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 8 | RSCAN0GAFLP1_8 | 00000000 _H | 32 | FFD2058C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 9 | RSCAN0GAFLID9 | 00000000 _H | 32 | FFD20590 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 9 | RSCAN0GAFLM9 | 00000000 _H | 32 | FFD20594 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 9 | RSCAN0GAFLP0_9 | 00000000 _H | 32 | FFD20598 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 9 | RSCAN0GAFLP1_9 | 00000000 _H | 32 | FFD2059C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 10 | RSCAN0GAFLID10 | 00000000 _H | 32 | FFD205A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 10 | RSCAN0GAFLM10 | 00000000 _H | 32 | FFD205A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 10 | RSCAN0GAFLP0_10 | 00000000 _H | 32 | FFD205A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 10 | RSCAN0GAFLP1_10 | 00000000 _H | 32 | FFD205AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 11 | RSCAN0GAFLID11 | 00000000 _H | 32 | FFD205B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 11 | RSCAN0GAFLM11 | 00000000 _H | 32 | FFD205B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 11 | RSCAN0GAFLP0_11 | 00000000 _H | 32 | FFD205B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 11 | RSCAN0GAFLP1_11 | 00000000 _H | 32 | FFD205BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 12 | RSCAN0GAFLID12 | 00000000 _H | 32 | FFD205C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 12 | RSCAN0GAFLM12 | 00000000 _H | 32 | FFD205C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 12 | RSCAN0GAFLP0_12 | 00000000 _H | 32 | FFD205C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 12 | RSCAN0GAFLP1_12 | 00000000 _H | 32 | FFD205CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 13 | RSCAN0GAFLID13 | 00000000 _H | 32 | FFD205D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 13 | RSCAN0GAFLM13 | 00000000 _H | 32 | FFD205D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 13 | RSCAN0GAFLP0_13 | 00000000 _H | 32 | FFD205D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 13 | RSCAN0GAFLP1_13 | 00000000 _H | 32 | FFD205DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule ID register 14 | RSCAN0GAFLID14 | 00000000 _H | 32 | FFD205E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule mask register 14 | RSCAN0GAFLM14 | 00000000 _H | 32 | FFD205E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 0 register 14 | RSCAN0GAFLP0_14 | 00000000 _H | 32 | FFD205E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive rule pointer 1 register 14 | RSCAN0GAFLP1_14 | 00000000 _H | 32 | FFD205EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (33/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Receive rule ID register 15 | RSCAN0GAFLID15 | 00000000 _H | 32 | FFD205F0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive rule mask register 15 | RSCAN0GAFLM15 | 00000000 _H | 32 | FFD205F4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive rule pointer 0 register 15 | RSCAN0GAFLP0_15 | 00000000 _H | 32 | FFD205F8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive rule pointer 1 register 15 | RSCAN0GAFLP1_15 | 00000000 _H | 32 | FFD205FC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 0 | RSCAN0RMID0 | 00000000 _H | 32 | FFD20600 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 0 | RSCAN0RMPTR0 | 00000000 _H | 32 | FFD20604 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 0 | RSCAN0RMDf0_0 | 00000000 _H | 32 | FFD20608 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 0 | RSCAN0RMDf1_0 | 00000000 _H | 32 | FFD2060C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 1 | RSCAN0RMID1 | 00000000 _H | 32 | FFD20610 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 1 | RSCAN0RMPTR1 | 00000000 _H | 32 | FFD20614 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 1 | RSCAN0RMDf0_1 | 00000000 _H | 32 | FFD20618 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 1 | RSCAN0RMDf1_1 | 00000000 _H | 32 | FFD2061C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 2 | RSCAN0RMID2 | 00000000 _H | 32 | FFD20620 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 2 | RSCAN0RMPTR2 | 00000000 _H | 32 | FFD20624 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 2 | RSCAN0RMDf0_2 | 00000000 _H | 32 | FFD20628 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 2 | RSCAN0RMDf1_2 | 00000000 _H | 32 | FFD2062C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 3 | RSCAN0RMID3 | 00000000 _H | 32 | FFD20630 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 3 | RSCAN0RMPTR3 | 00000000 _H | 32 | FFD20634 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 3 | RSCAN0RMDf0_3 | 00000000 _H | 32 | FFD20638 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 3 | RSCAN0RMDf1_3 | 00000000 _H | 32 | FFD2063C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 4 | RSCAN0RMID4 | 00000000 _H | 32 | FFD20640 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 4 | RSCAN0RMPTR4 | 00000000 _H | 32 | FFD20644 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 4 | RSCAN0RMDf0_4 | 00000000 _H | 32 | FFD20648 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 4 | RSCAN0RMDf1_4 | 00000000 _H | 32 | FFD2064C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 5 | RSCAN0RMID5 | 00000000 _H | 32 | FFD20650 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 5 | RSCAN0RMPTR5 | 00000000 _H | 32 | FFD20654 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 5 | RSCAN0RMDf0_5 | 00000000 _H | 32 | FFD20658 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 5 | RSCAN0RMDf1_5 | 00000000 _H | 32 | FFD2065C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 6 | RSCAN0RMID6 | 00000000 _H | 32 | FFD20660 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 6 | RSCAN0RMPTR6 | 00000000 _H | 32 | FFD20664 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 6 | RSCAN0RMDf0_6 | 00000000 _H | 32 | FFD20668 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 6 | RSCAN0RMDf1_6 | 00000000 _H | 32 | FFD2066C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 7 | RSCAN0RMID7 | 00000000 _H | 32 | FFD20670 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 7 | RSCAN0RMPTR7 | 00000000 _H | 32 | FFD20674 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 7 | RSCAN0RMDf0_7 | 00000000 _H | 32 | FFD20678 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 7 | RSCAN0RMDf1_7 | 00000000 _H | 32 | FFD2067C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 8 | RSCAN0RMID8 | 00000000 _H | 32 | FFD20680 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 8 | RSCAN0RMPTR8 | 00000000 _H | 32 | FFD20684 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 8 | RSCAN0RMDf0_8 | 00000000 _H | 32 | FFD20688 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 8 | RSCAN0RMDf1_8 | 00000000 _H | 32 | FFD2068C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 9 | RSCAN0RMID9 | 00000000 _H | 32 | FFD20690 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 9 | RSCAN0RMPTR9 | 00000000 _H | 32 | FFD20694 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 9 | RSCAN0RMDf0_9 | 00000000 _H | 32 | FFD20698 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 9 | RSCAN0RMDf1_9 | 00000000 _H | 32 | FFD2069C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 10 | RSCAN0RMID10 | 00000000 _H | 32 | FFD206A0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 10 | RSCAN0RMPTR10 | 00000000 _H | 32 | FFD206A4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 10 | RSCAN0RMDf0_10 | 00000000 _H | 32 | FFD206A8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 10 | RSCAN0RMDf1_10 | 00000000 _H | 32 | FFD206AC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (34/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Receive buffer ID register 11 | RSCAN0RMID11 | 00000000 _H | 32 | FFD206B0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 11 | RSCAN0RMPTR11 | 00000000 _H | 32 | FFD206B4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 11 | RSCAN0RMDf0_11 | 00000000 _H | 32 | FFD206B8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 11 | RSCAN0RMDf1_11 | 00000000 _H | 32 | FFD206BC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 12 | RSCAN0RMID12 | 00000000 _H | 32 | FFD206C0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 12 | RSCAN0RMPTR12 | 00000000 _H | 32 | FFD206C4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 12 | RSCAN0RMDf0_12 | 00000000 _H | 32 | FFD206C8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 12 | RSCAN0RMDf1_12 | 00000000 _H | 32 | FFD206CC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 13 | RSCAN0RMID13 | 00000000 _H | 32 | FFD206D0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 13 | RSCAN0RMPTR13 | 00000000 _H | 32 | FFD206D4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 13 | RSCAN0RMDf0_13 | 00000000 _H | 32 | FFD206D8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 13 | RSCAN0RMDf1_13 | 00000000 _H | 32 | FFD206DC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 14 | RSCAN0RMID14 | 00000000 _H | 32 | FFD206E0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 14 | RSCAN0RMPTR14 | 00000000 _H | 32 | FFD206E4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 14 | RSCAN0RMDf0_14 | 00000000 _H | 32 | FFD206E8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 14 | RSCAN0RMDf1_14 | 00000000 _H | 32 | FFD206EC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 15 | RSCAN0RMID15 | 00000000 _H | 32 | FFD206F0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 15 | RSCAN0RMPTR15 | 00000000 _H | 32 | FFD206F4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 15 | RSCAN0RMDf0_15 | 00000000 _H | 32 | FFD206F8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 15 | RSCAN0RMDf1_15 | 00000000 _H | 32 | FFD206FC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 16 | RSCAN0RMID16 | 00000000 _H | 32 | FFD20700 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 16 | RSCAN0RMPTR16 | 00000000 _H | 32 | FFD20704 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 16 | RSCAN0RMDf0_16 | 00000000 _H | 32 | FFD20708 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 16 | RSCAN0RMDf1_16 | 00000000 _H | 32 | FFD2070C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 17 | RSCAN0RMID17 | 00000000 _H | 32 | FFD20710 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 17 | RSCAN0RMPTR17 | 00000000 _H | 32 | FFD20714 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 17 | RSCAN0RMDf0_17 | 00000000 _H | 32 | FFD20718 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 17 | RSCAN0RMDf1_17 | 00000000 _H | 32 | FFD2071C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 18 | RSCAN0RMID18 | 00000000 _H | 32 | FFD20720 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 18 | RSCAN0RMPTR18 | 00000000 _H | 32 | FFD20724 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 18 | RSCAN0RMDf0_18 | 00000000 _H | 32 | FFD20728 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 18 | RSCAN0RMDf1_18 | 00000000 _H | 32 | FFD2072C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 19 | RSCAN0RMID19 | 00000000 _H | 32 | FFD20730 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 19 | RSCAN0RMPTR19 | 00000000 _H | 32 | FFD20734 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 19 | RSCAN0RMDf0_19 | 00000000 _H | 32 | FFD20738 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 19 | RSCAN0RMDf1_19 | 00000000 _H | 32 | FFD2073C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 20 | RSCAN0RMID20 | 00000000 _H | 32 | FFD20740 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 20 | RSCAN0RMPTR20 | 00000000 _H | 32 | FFD20744 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 20 | RSCAN0RMDf0_20 | 00000000 _H | 32 | FFD20748 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 20 | RSCAN0RMDf1_20 | 00000000 _H | 32 | FFD2074C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 21 | RSCAN0RMID21 | 00000000 _H | 32 | FFD20750 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 21 | RSCAN0RMPTR21 | 00000000 _H | 32 | FFD20754 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 21 | RSCAN0RMDf0_21 | 00000000 _H | 32 | FFD20758 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 21 | RSCAN0RMDf1_21 | 00000000 _H | 32 | FFD2075C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 22 | RSCAN0RMID22 | 00000000 _H | 32 | FFD20760 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 22 | RSCAN0RMPTR22 | 00000000 _H | 32 | FFD20764 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 22 | RSCAN0RMDf0_22 | 00000000 _H | 32 | FFD20768 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 22 | RSCAN0RMDf1_22 | 00000000 _H | 32 | FFD2076C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (35/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Receive buffer ID register 23 | RSCAN0RMID23 | 00000000 _H | 32 | FFD20770 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 23 | RSCAN0RMPTR23 | 00000000 _H | 32 | FFD20774 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 23 | RSCAN0RMDf0_23 | 00000000 _H | 32 | FFD20778 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 23 | RSCAN0RMDf1_23 | 00000000 _H | 32 | FFD2077C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 24 | RSCAN0RMID24 | 00000000 _H | 32 | FFD20780 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 24 | RSCAN0RMPTR24 | 00000000 _H | 32 | FFD20784 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 24 | RSCAN0RMDf0_24 | 00000000 _H | 32 | FFD20788 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 24 | RSCAN0RMDf1_24 | 00000000 _H | 32 | FFD2078C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 25 | RSCAN0RMID25 | 00000000 _H | 32 | FFD20790 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 25 | RSCAN0RMPTR25 | 00000000 _H | 32 | FFD20794 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 25 | RSCAN0RMDf0_25 | 00000000 _H | 32 | FFD20798 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 25 | RSCAN0RMDf1_25 | 00000000 _H | 32 | FFD2079C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 26 | RSCAN0RMID26 | 00000000 _H | 32 | FFD207A0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 26 | RSCAN0RMPTR26 | 00000000 _H | 32 | FFD207A4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 26 | RSCAN0RMDf0_26 | 00000000 _H | 32 | FFD207A8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 26 | RSCAN0RMDf1_26 | 00000000 _H | 32 | FFD207AC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 27 | RSCAN0RMID27 | 00000000 _H | 32 | FFD207B0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 27 | RSCAN0RMPTR27 | 00000000 _H | 32 | FFD207B4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 27 | RSCAN0RMDf0_27 | 00000000 _H | 32 | FFD207B8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 27 | RSCAN0RMDf1_27 | 00000000 _H | 32 | FFD207BC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 28 | RSCAN0RMID28 | 00000000 _H | 32 | FFD207C0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 28 | RSCAN0RMPTR28 | 00000000 _H | 32 | FFD207C4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 28 | RSCAN0RMDf0_28 | 00000000 _H | 32 | FFD207C8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 28 | RSCAN0RMDf1_28 | 00000000 _H | 32 | FFD207CC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 29 | RSCAN0RMID29 | 00000000 _H | 32 | FFD207D0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 29 | RSCAN0RMPTR29 | 00000000 _H | 32 | FFD207D4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 29 | RSCAN0RMDf0_29 | 00000000 _H | 32 | FFD207D8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 29 | RSCAN0RMDf1_29 | 00000000 _H | 32 | FFD207DC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 30 | RSCAN0RMID30 | 00000000 _H | 32 | FFD207E0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 30 | RSCAN0RMPTR30 | 00000000 _H | 32 | FFD207E4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 30 | RSCAN0RMDf0_30 | 00000000 _H | 32 | FFD207E8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 30 | RSCAN0RMDf1_30 | 00000000 _H | 32 | FFD207EC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 31 | RSCAN0RMID31 | 00000000 _H | 32 | FFD207F0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 31 | RSCAN0RMPTR31 | 00000000 _H | 32 | FFD207F4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 31 | RSCAN0RMDf0_31 | 00000000 _H | 32 | FFD207F8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 31 | RSCAN0RMDf1_31 | 00000000 _H | 32 | FFD207FC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 32 | RSCAN0RMID32 | 00000000 _H | 32 | FFD20800 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 32 | RSCAN0RMPTR32 | 00000000 _H | 32 | FFD20804 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 32 | RSCAN0RMDf0_32 | 00000000 _H | 32 | FFD20808 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 32 | RSCAN0RMDf1_32 | 00000000 _H | 32 | FFD2080C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 33 | RSCAN0RMID33 | 00000000 _H | 32 | FFD20810 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 33 | RSCAN0RMPTR33 | 00000000 _H | 32 | FFD20814 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 33 | RSCAN0RMDf0_33 | 00000000 _H | 32 | FFD20818 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 33 | RSCAN0RMDf1_33 | 00000000 _H | 32 | FFD2081C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer ID register 34 | RSCAN0RMID34 | 00000000 _H | 32 | FFD20820 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer pointer register 34 | RSCAN0RMPTR34 | 00000000 _H | 32 | FFD20824 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 0 register 34 | RSCAN0RMDf0_34 | 00000000 _H | 32 | FFD20828 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCAN0 | Receive buffer data field 1 register 34 | RSCAN0RMDf1_34 | 00000000 _H | 32 | FFD2082C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (36/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Receive buffer ID register 35 | RSCAN0RMID35 | 00000000 _H | 32 | FFD20830 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 35 | RSCAN0RMPTR35 | 00000000 _H | 32 | FFD20834 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 35 | RSCAN0RMDf0_35 | 00000000 _H | 32 | FFD20838 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 35 | RSCAN0RMDf1_35 | 00000000 _H | 32 | FFD2083C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 36 | RSCAN0RMID36 | 00000000 _H | 32 | FFD20840 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 36 | RSCAN0RMPTR36 | 00000000 _H | 32 | FFD20844 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 36 | RSCAN0RMDf0_36 | 00000000 _H | 32 | FFD20848 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 36 | RSCAN0RMDf1_36 | 00000000 _H | 32 | FFD2084C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 37 | RSCAN0RMID37 | 00000000 _H | 32 | FFD20850 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 37 | RSCAN0RMPTR37 | 00000000 _H | 32 | FFD20854 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 37 | RSCAN0RMDf0_37 | 00000000 _H | 32 | FFD20858 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 37 | RSCAN0RMDf1_37 | 00000000 _H | 32 | FFD2085C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 38 | RSCAN0RMID38 | 00000000 _H | 32 | FFD20860 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 38 | RSCAN0RMPTR38 | 00000000 _H | 32 | FFD20864 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 38 | RSCAN0RMDf0_38 | 00000000 _H | 32 | FFD20868 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 38 | RSCAN0RMDf1_38 | 00000000 _H | 32 | FFD2086C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 39 | RSCAN0RMID39 | 00000000 _H | 32 | FFD20870 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 39 | RSCAN0RMPTR39 | 00000000 _H | 32 | FFD20874 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 39 | RSCAN0RMDf0_39 | 00000000 _H | 32 | FFD20878 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 39 | RSCAN0RMDf1_39 | 00000000 _H | 32 | FFD2087C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 40 | RSCAN0RMID40 | 00000000 _H | 32 | FFD20880 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 40 | RSCAN0RMPTR40 | 00000000 _H | 32 | FFD20884 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 40 | RSCAN0RMDf0_40 | 00000000 _H | 32 | FFD20888 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 40 | RSCAN0RMDf1_40 | 00000000 _H | 32 | FFD2088C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 41 | RSCAN0RMID41 | 00000000 _H | 32 | FFD20890 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 41 | RSCAN0RMPTR41 | 00000000 _H | 32 | FFD20894 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 41 | RSCAN0RMDf0_41 | 00000000 _H | 32 | FFD20898 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 41 | RSCAN0RMDf1_41 | 00000000 _H | 32 | FFD2089C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 42 | RSCAN0RMID42 | 00000000 _H | 32 | FFD208A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 42 | RSCAN0RMPTR42 | 00000000 _H | 32 | FFD208A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 42 | RSCAN0RMDf0_42 | 00000000 _H | 32 | FFD208A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 42 | RSCAN0RMDf1_42 | 00000000 _H | 32 | FFD208AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 43 | RSCAN0RMID43 | 00000000 _H | 32 | FFD208B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 43 | RSCAN0RMPTR43 | 00000000 _H | 32 | FFD208B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 43 | RSCAN0RMDf0_43 | 00000000 _H | 32 | FFD208B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 43 | RSCAN0RMDf1_43 | 00000000 _H | 32 | FFD208BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 44 | RSCAN0RMID44 | 00000000 _H | 32 | FFD208C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 44 | RSCAN0RMPTR44 | 00000000 _H | 32 | FFD208C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 44 | RSCAN0RMDf0_44 | 00000000 _H | 32 | FFD208C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 44 | RSCAN0RMDf1_44 | 00000000 _H | 32 | FFD208CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 45 | RSCAN0RMID45 | 00000000 _H | 32 | FFD208D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 45 | RSCAN0RMPTR45 | 00000000 _H | 32 | FFD208D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 45 | RSCAN0RMDf0_45 | 00000000 _H | 32 | FFD208D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 45 | RSCAN0RMDf1_45 | 00000000 _H | 32 | FFD208DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer ID register 46 | RSCAN0RMID46 | 00000000 _H | 32 | FFD208E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 46 | RSCAN0RMPTR46 | 00000000 _H | 32 | FFD208E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 46 | RSCAN0RMDf0_46 | 00000000 _H | 32 | FFD208E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 46 | RSCAN0RMDf1_46 | 00000000 _H | 32 | FFD208EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (37/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Receive buffer ID register 47 | RSCAN0RMID47 | 00000000 _H | 32 | FFD208F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer pointer register 47 | RSCAN0RMPTR47 | 00000000 _H | 32 | FFD208F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 0 register 47 | RSCAN0RMDf0_47 | 00000000 _H | 32 | FFD208F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive buffer data field 1 register 47 | RSCAN0RMDf1_47 | 00000000 _H | 32 | FFD208FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 0 | RSCAN0RFID0 | 00000000 _H | 32 | FFD20E00 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 0 | RSCAN0RFPTR0 | 00000000 _H | 32 | FFD20E04 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 0 | RSCAN0RFDf0_0 | 00000000 _H | 32 | FFD20E08 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 0 | RSCAN0RFDf1_0 | 00000000 _H | 32 | FFD20E0C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 1 | RSCAN0RFID1 | 00000000 _H | 32 | FFD20E10 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 1 | RSCAN0RFPTR1 | 00000000 _H | 32 | FFD20E14 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 1 | RSCAN0RFDf0_1 | 00000000 _H | 32 | FFD20E18 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 1 | RSCAN0RFDf1_1 | 00000000 _H | 32 | FFD20E1C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 2 | RSCAN0RFID2 | 00000000 _H | 32 | FFD20E20 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 2 | RSCAN0RFPTR2 | 00000000 _H | 32 | FFD20E24 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 2 | RSCAN0RFDf0_2 | 00000000 _H | 32 | FFD20E28 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 2 | RSCAN0RFDf1_2 | 00000000 _H | 32 | FFD20E2C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 3 | RSCAN0RFID3 | 00000000 _H | 32 | FFD20E30 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 3 | RSCAN0RFPTR3 | 00000000 _H | 32 | FFD20E34 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 3 | RSCAN0RFDf0_3 | 00000000 _H | 32 | FFD20E38 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 3 | RSCAN0RFDf1_3 | 00000000 _H | 32 | FFD20E3C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 4 | RSCAN0RFID4 | 00000000 _H | 32 | FFD20E40 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 4 | RSCAN0RFPTR4 | 00000000 _H | 32 | FFD20E44 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 4 | RSCAN0RFDf0_4 | 00000000 _H | 32 | FFD20E48 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 4 | RSCAN0RFDf1_4 | 00000000 _H | 32 | FFD20E4C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 5 | RSCAN0RFID5 | 00000000 _H | 32 | FFD20E50 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 5 | RSCAN0RFPTR5 | 00000000 _H | 32 | FFD20E54 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 5 | RSCAN0RFDf0_5 | 00000000 _H | 32 | FFD20E58 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 5 | RSCAN0RFDf1_5 | 00000000 _H | 32 | FFD20E5C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 6 | RSCAN0RFID6 | 00000000 _H | 32 | FFD20E60 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 6 | RSCAN0RFPTR6 | 00000000 _H | 32 | FFD20E64 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 6 | RSCAN0RFDf0_6 | 00000000 _H | 32 | FFD20E68 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 6 | RSCAN0RFDf1_6 | 00000000 _H | 32 | FFD20E6C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access ID register 7 | RSCAN0RFID7 | 00000000 _H | 32 | FFD20E70 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access pointer register 7 | RSCAN0RFPTR7 | 00000000 _H | 32 | FFD20E74 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 0 register 7 | RSCAN0RFDf0_7 | 00000000 _H | 32 | FFD20E78 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Receive FIFO buffer access data field 1 register 7 | RSCAN0RFDf1_7 | 00000000 _H | 32 | FFD20E7C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 0 | RSCAN0CFID0 | 00000000 _H | 32 | FFD20E80 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 0 | RSCAN0CFPTR0 | 00000000 _H | 32 | FFD20E84 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 0 | RSCAN0CFDf0_0 | 00000000 _H | 32 | FFD20E88 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 0 | RSCAN0CFDf1_0 | 00000000 _H | 32 | FFD20E8C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 1 | RSCAN0CFID1 | 00000000 _H | 32 | FFD20E90 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 1 | RSCAN0CFPTR1 | 00000000 _H | 32 | FFD20E94 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 1 | RSCAN0CFDf0_1 | 00000000 _H | 32 | FFD20E98 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 1 | RSCAN0CFDf1_1 | 00000000 _H | 32 | FFD20E9C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 2 | RSCAN0CFID2 | 00000000 _H | 32 | FFD20EA0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (38/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 2 | RSCAN0CFPTR2 | 00000000 _H | 32 | FFD20EA4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 2 | RSCAN0CFDF0_2 | 00000000 _H | 32 | FFD20EA8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 2 | RSCAN0CFDF1_2 | 00000000 _H | 32 | FFD20EAC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 3 | RSCAN0CFID3 | 00000000 _H | 32 | FFD20EB0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 3 | RSCAN0CFPTR3 | 00000000 _H | 32 | FFD20EB4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 3 | RSCAN0CFDF0_3 | 00000000 _H | 32 | FFD20EB8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 3 | RSCAN0CFDF1_3 | 00000000 _H | 32 | FFD20EBC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 4 | RSCAN0CFID4 | 00000000 _H | 32 | FFD20EC0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 4 | RSCAN0CFPTR4 | 00000000 _H | 32 | FFD20EC4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 4 | RSCAN0CFDF0_4 | 00000000 _H | 32 | FFD20EC8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 4 | RSCAN0CFDF1_4 | 00000000 _H | 32 | FFD20ECC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 5 | RSCAN0CFID5 | 00000000 _H | 32 | FFD20ED0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 5 | RSCAN0CFPTR5 | 00000000 _H | 32 | FFD20ED4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 5 | RSCAN0CFDF0_5 | 00000000 _H | 32 | FFD20ED8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 5 | RSCAN0CFDF1_5 | 00000000 _H | 32 | FFD20EDC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 6 | RSCAN0CFID6 | 00000000 _H | 32 | FFD20EE0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 6 | RSCAN0CFPTR6 | 00000000 _H | 32 | FFD20EE4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 6 | RSCAN0CFDF0_6 | 00000000 _H | 32 | FFD20EE8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 6 | RSCAN0CFDF1_6 | 00000000 _H | 32 | FFD20EEC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 7 | RSCAN0CFID7 | 00000000 _H | 32 | FFD20EF0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 7 | RSCAN0CFPTR7 | 00000000 _H | 32 | FFD20EF4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 7 | RSCAN0CFDF0_7 | 00000000 _H | 32 | FFD20EF8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 7 | RSCAN0CFDF1_7 | 00000000 _H | 32 | FFD20EFC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access ID register 8 | RSCAN0CFID8 | 00000000 _H | 32 | FFD20F00 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access pointer register 8 | RSCAN0CFPTR8 | 00000000 _H | 32 | FFD20F04 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 0 register 8 | RSCAN0CFDF0_8 | 00000000 _H | 32 | FFD20F08 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit/receive FIFO buffer access data field 1 register 8 | RSCAN0CFDF1_8 | 00000000 _H | 32 | FFD20F0C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 0 | RSCAN0TMID0 | 00000000 _H | 32 | FFD21000 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 0 | RSCAN0TMPTR0 | 00000000 _H | 32 | FFD21004 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 0 | RSCAN0TMDF0_0 | 00000000 _H | 32 | FFD21008 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 0 | RSCAN0TMDF1_0 | 00000000 _H | 32 | FFD2100C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 1 | RSCAN0TMID1 | 00000000 _H | 32 | FFD21010 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 1 | RSCAN0TMPTR1 | 00000000 _H | 32 | FFD21014 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 1 | RSCAN0TMDF0_1 | 00000000 _H | 32 | FFD21018 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 1 | RSCAN0TMDF1_1 | 00000000 _H | 32 | FFD2101C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 2 | RSCAN0TMID2 | 00000000 _H | 32 | FFD21020 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 2 | RSCAN0TMPTR2 | 00000000 _H | 32 | FFD21024 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 2 | RSCAN0TMDF0_2 | 00000000 _H | 32 | FFD21028 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 2 | RSCAN0TMDF1_2 | 00000000 _H | 32 | FFD2102C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 3 | RSCAN0TMID3 | 00000000 _H | 32 | FFD21030 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 3 | RSCAN0TMPTR3 | 00000000 _H | 32 | FFD21034 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (39/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit buffer data field 0 register 3 | RSCAN0TMDf0_3 | 00000000 _H | 32 | FFD21038 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 3 | RSCAN0TMDf1_3 | 00000000 _H | 32 | FFD2103C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 4 | RSCAN0TMID4 | 00000000 _H | 32 | FFD21040 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 4 | RSCAN0TMPTR4 | 00000000 _H | 32 | FFD21044 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 4 | RSCAN0TMDf0_4 | 00000000 _H | 32 | FFD21048 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 4 | RSCAN0TMDf1_4 | 00000000 _H | 32 | FFD2104C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 5 | RSCAN0TMID5 | 00000000 _H | 32 | FFD21050 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 5 | RSCAN0TMPTR5 | 00000000 _H | 32 | FFD21054 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 5 | RSCAN0TMDf0_5 | 00000000 _H | 32 | FFD21058 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 5 | RSCAN0TMDf1_5 | 00000000 _H | 32 | FFD2105C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 6 | RSCAN0TMID6 | 00000000 _H | 32 | FFD21060 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 6 | RSCAN0TMPTR6 | 00000000 _H | 32 | FFD21064 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 6 | RSCAN0TMDf0_6 | 00000000 _H | 32 | FFD21068 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 6 | RSCAN0TMDf1_6 | 00000000 _H | 32 | FFD2106C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 7 | RSCAN0TMID7 | 00000000 _H | 32 | FFD21070 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 7 | RSCAN0TMPTR7 | 00000000 _H | 32 | FFD21074 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 7 | RSCAN0TMDf0_7 | 00000000 _H | 32 | FFD21078 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 7 | RSCAN0TMDf1_7 | 00000000 _H | 32 | FFD2107C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 8 | RSCAN0TMID8 | 00000000 _H | 32 | FFD21080 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 8 | RSCAN0TMPTR8 | 00000000 _H | 32 | FFD21084 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 8 | RSCAN0TMDf0_8 | 00000000 _H | 32 | FFD21088 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 8 | RSCAN0TMDf1_8 | 00000000 _H | 32 | FFD2108C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 9 | RSCAN0TMID9 | 00000000 _H | 32 | FFD21090 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 9 | RSCAN0TMPTR9 | 00000000 _H | 32 | FFD21094 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 9 | RSCAN0TMDf0_9 | 00000000 _H | 32 | FFD21098 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 9 | RSCAN0TMDf1_9 | 00000000 _H | 32 | FFD2109C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 10 | RSCAN0TMID10 | 00000000 _H | 32 | FFD210A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 10 | RSCAN0TMPTR10 | 00000000 _H | 32 | FFD210A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 10 | RSCAN0TMDf0_10 | 00000000 _H | 32 | FFD210A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 10 | RSCAN0TMDf1_10 | 00000000 _H | 32 | FFD210AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 11 | RSCAN0TMID11 | 00000000 _H | 32 | FFD210B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 11 | RSCAN0TMPTR11 | 00000000 _H | 32 | FFD210B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 11 | RSCAN0TMDf0_11 | 00000000 _H | 32 | FFD210B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 11 | RSCAN0TMDf1_11 | 00000000 _H | 32 | FFD210BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 12 | RSCAN0TMID12 | 00000000 _H | 32 | FFD210C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 12 | RSCAN0TMPTR12 | 00000000 _H | 32 | FFD210C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 12 | RSCAN0TMDf0_12 | 00000000 _H | 32 | FFD210C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 12 | RSCAN0TMDf1_12 | 00000000 _H | 32 | FFD210CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 13 | RSCAN0TMID13 | 00000000 _H | 32 | FFD210D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 13 | RSCAN0TMPTR13 | 00000000 _H | 32 | FFD210D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 13 | RSCAN0TMDf0_13 | 00000000 _H | 32 | FFD210D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 13 | RSCAN0TMDf1_13 | 00000000 _H | 32 | FFD210DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 14 | RSCAN0TMID14 | 00000000 _H | 32 | FFD210E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 14 | RSCAN0TMPTR14 | 00000000 _H | 32 | FFD210E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 14 | RSCAN0TMDf0_14 | 00000000 _H | 32 | FFD210E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 14 | RSCAN0TMDf1_14 | 00000000 _H | 32 | FFD210EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 15 | RSCAN0TMID15 | 00000000 _H | 32 | FFD210F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 15 | RSCAN0TMPTR15 | 00000000 _H | 32 | FFD210F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (40/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit buffer data field 0 register 15 | RSCAN0TMDf0_15 | 00000000 _H | 32 | FFD210F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 15 | RSCAN0TMDf1_15 | 00000000 _H | 32 | FFD210FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 16 | RSCAN0TMD16 | 00000000 _H | 32 | FFD21100 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 16 | RSCAN0TMPTR16 | 00000000 _H | 32 | FFD21104 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 16 | RSCAN0TMDf0_16 | 00000000 _H | 32 | FFD21108 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 16 | RSCAN0TMDf1_16 | 00000000 _H | 32 | FFD2110C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 17 | RSCAN0TMD17 | 00000000 _H | 32 | FFD21110 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 17 | RSCAN0TMPTR17 | 00000000 _H | 32 | FFD21114 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 17 | RSCAN0TMDf0_17 | 00000000 _H | 32 | FFD21118 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 17 | RSCAN0TMDf1_17 | 00000000 _H | 32 | FFD2111C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 18 | RSCAN0TMD18 | 00000000 _H | 32 | FFD21120 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 18 | RSCAN0TMPTR18 | 00000000 _H | 32 | FFD21124 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 18 | RSCAN0TMDf0_18 | 00000000 _H | 32 | FFD21128 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 18 | RSCAN0TMDf1_18 | 00000000 _H | 32 | FFD2112C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 19 | RSCAN0TMD19 | 00000000 _H | 32 | FFD21130 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 19 | RSCAN0TMPTR19 | 00000000 _H | 32 | FFD21134 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 19 | RSCAN0TMDf0_19 | 00000000 _H | 32 | FFD21138 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 19 | RSCAN0TMDf1_19 | 00000000 _H | 32 | FFD2113C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 20 | RSCAN0TMD20 | 00000000 _H | 32 | FFD21140 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 20 | RSCAN0TMPTR20 | 00000000 _H | 32 | FFD21144 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 20 | RSCAN0TMDf0_20 | 00000000 _H | 32 | FFD21148 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 20 | RSCAN0TMDf1_20 | 00000000 _H | 32 | FFD2114C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 21 | RSCAN0TMD21 | 00000000 _H | 32 | FFD21150 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 21 | RSCAN0TMPTR21 | 00000000 _H | 32 | FFD21154 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 21 | RSCAN0TMDf0_21 | 00000000 _H | 32 | FFD21158 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 21 | RSCAN0TMDf1_21 | 00000000 _H | 32 | FFD2115C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 22 | RSCAN0TMD22 | 00000000 _H | 32 | FFD21160 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 22 | RSCAN0TMPTR22 | 00000000 _H | 32 | FFD21164 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 22 | RSCAN0TMDf0_22 | 00000000 _H | 32 | FFD21168 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 22 | RSCAN0TMDf1_22 | 00000000 _H | 32 | FFD2116C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 23 | RSCAN0TMD23 | 00000000 _H | 32 | FFD21170 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 23 | RSCAN0TMPTR23 | 00000000 _H | 32 | FFD21174 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 23 | RSCAN0TMDf0_23 | 00000000 _H | 32 | FFD21178 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 23 | RSCAN0TMDf1_23 | 00000000 _H | 32 | FFD2117C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 24 | RSCAN0TMD24 | 00000000 _H | 32 | FFD21180 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 24 | RSCAN0TMPTR24 | 00000000 _H | 32 | FFD21184 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 24 | RSCAN0TMDf0_24 | 00000000 _H | 32 | FFD21188 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 24 | RSCAN0TMDf1_24 | 00000000 _H | 32 | FFD2118C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 25 | RSCAN0TMD25 | 00000000 _H | 32 | FFD21190 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 25 | RSCAN0TMPTR25 | 00000000 _H | 32 | FFD21194 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 25 | RSCAN0TMDf0_25 | 00000000 _H | 32 | FFD21198 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 25 | RSCAN0TMDf1_25 | 00000000 _H | 32 | FFD2119C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 26 | RSCAN0TMD26 | 00000000 _H | 32 | FFD211A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 26 | RSCAN0TMPTR26 | 00000000 _H | 32 | FFD211A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 26 | RSCAN0TMDf0_26 | 00000000 _H | 32 | FFD211A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 26 | RSCAN0TMDf1_26 | 00000000 _H | 32 | FFD211AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 27 | RSCAN0TMD27 | 00000000 _H | 32 | FFD211B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 27 | RSCAN0TMPTR27 | 00000000 _H | 32 | FFD211B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (41/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit buffer data field 0 register 27 | RSCAN0TMDf0_27 | 00000000 _H | 32 | FFD211B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 27 | RSCAN0TMDf1_27 | 00000000 _H | 32 | FFD211BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 28 | RSCAN0TMD28 | 00000000 _H | 32 | FFD211C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 28 | RSCAN0TMPTR28 | 00000000 _H | 32 | FFD211C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 28 | RSCAN0TMDf0_28 | 00000000 _H | 32 | FFD211C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 28 | RSCAN0TMDf1_28 | 00000000 _H | 32 | FFD211CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 29 | RSCAN0TMD29 | 00000000 _H | 32 | FFD211D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 29 | RSCAN0TMPTR29 | 00000000 _H | 32 | FFD211D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 29 | RSCAN0TMDf0_29 | 00000000 _H | 32 | FFD211D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 29 | RSCAN0TMDf1_29 | 00000000 _H | 32 | FFD211DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 30 | RSCAN0TMD30 | 00000000 _H | 32 | FFD211E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 30 | RSCAN0TMPTR30 | 00000000 _H | 32 | FFD211E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 30 | RSCAN0TMDf0_30 | 00000000 _H | 32 | FFD211E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 30 | RSCAN0TMDf1_30 | 00000000 _H | 32 | FFD211EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 31 | RSCAN0TMD31 | 00000000 _H | 32 | FFD211F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 31 | RSCAN0TMPTR31 | 00000000 _H | 32 | FFD211F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 31 | RSCAN0TMDf0_31 | 00000000 _H | 32 | FFD211F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 31 | RSCAN0TMDf1_31 | 00000000 _H | 32 | FFD211FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 32 | RSCAN0TMD32 | 00000000 _H | 32 | FFD21200 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 32 | RSCAN0TMPTR32 | 00000000 _H | 32 | FFD21204 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 32 | RSCAN0TMDf0_32 | 00000000 _H | 32 | FFD21208 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 32 | RSCAN0TMDf1_32 | 00000000 _H | 32 | FFD2120C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 33 | RSCAN0TMD33 | 00000000 _H | 32 | FFD21210 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 33 | RSCAN0TMPTR33 | 00000000 _H | 32 | FFD21214 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 33 | RSCAN0TMDf0_33 | 00000000 _H | 32 | FFD21218 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 33 | RSCAN0TMDf1_33 | 00000000 _H | 32 | FFD2121C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 34 | RSCAN0TMD34 | 00000000 _H | 32 | FFD21220 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 34 | RSCAN0TMPTR34 | 00000000 _H | 32 | FFD21224 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 34 | RSCAN0TMDf0_34 | 00000000 _H | 32 | FFD21228 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 34 | RSCAN0TMDf1_34 | 00000000 _H | 32 | FFD2122C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 35 | RSCAN0TMD35 | 00000000 _H | 32 | FFD21230 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 35 | RSCAN0TMPTR35 | 00000000 _H | 32 | FFD21234 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 35 | RSCAN0TMDf0_35 | 00000000 _H | 32 | FFD21238 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 35 | RSCAN0TMDf1_35 | 00000000 _H | 32 | FFD2123C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 36 | RSCAN0TMD36 | 00000000 _H | 32 | FFD21240 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 36 | RSCAN0TMPTR36 | 00000000 _H | 32 | FFD21244 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 36 | RSCAN0TMDf0_36 | 00000000 _H | 32 | FFD21248 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 36 | RSCAN0TMDf1_36 | 00000000 _H | 32 | FFD2124C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 37 | RSCAN0TMD37 | 00000000 _H | 32 | FFD21250 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 37 | RSCAN0TMPTR37 | 00000000 _H | 32 | FFD21254 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 37 | RSCAN0TMDf0_37 | 00000000 _H | 32 | FFD21258 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 37 | RSCAN0TMDf1_37 | 00000000 _H | 32 | FFD2125C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 38 | RSCAN0TMD38 | 00000000 _H | 32 | FFD21260 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 38 | RSCAN0TMPTR38 | 00000000 _H | 32 | FFD21264 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 38 | RSCAN0TMDf0_38 | 00000000 _H | 32 | FFD21268 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 38 | RSCAN0TMDf1_38 | 00000000 _H | 32 | FFD2126C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 39 | RSCAN0TMD39 | 00000000 _H | 32 | FFD21270 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 39 | RSCAN0TMPTR39 | 00000000 _H | 32 | FFD21274 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (42/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | Transmit buffer data field 0 register 39 | RSCAN0TMDf0_39 | 00000000 _H | 32 | FFD21278 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 39 | RSCAN0TMDf1_39 | 00000000 _H | 32 | FFD2127C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 40 | RSCAN0TMD40 | 00000000 _H | 32 | FFD21280 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 40 | RSCAN0TMPTR40 | 00000000 _H | 32 | FFD21284 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 40 | RSCAN0TMDf0_40 | 00000000 _H | 32 | FFD21288 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 40 | RSCAN0TMDf1_40 | 00000000 _H | 32 | FFD2128C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 41 | RSCAN0TMD41 | 00000000 _H | 32 | FFD21290 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 41 | RSCAN0TMPTR41 | 00000000 _H | 32 | FFD21294 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 41 | RSCAN0TMDf0_41 | 00000000 _H | 32 | FFD21298 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 41 | RSCAN0TMDf1_41 | 00000000 _H | 32 | FFD2129C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 42 | RSCAN0TMD42 | 00000000 _H | 32 | FFD212A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 42 | RSCAN0TMPTR42 | 00000000 _H | 32 | FFD212A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 42 | RSCAN0TMDf0_42 | 00000000 _H | 32 | FFD212A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 42 | RSCAN0TMDf1_42 | 00000000 _H | 32 | FFD212AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 43 | RSCAN0TMD43 | 00000000 _H | 32 | FFD212B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 43 | RSCAN0TMPTR43 | 00000000 _H | 32 | FFD212B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 43 | RSCAN0TMDf0_43 | 00000000 _H | 32 | FFD212B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 43 | RSCAN0TMDf1_43 | 00000000 _H | 32 | FFD212BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 44 | RSCAN0TMD44 | 00000000 _H | 32 | FFD212C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 44 | RSCAN0TMPTR44 | 00000000 _H | 32 | FFD212C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 44 | RSCAN0TMDf0_44 | 00000000 _H | 32 | FFD212C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 44 | RSCAN0TMDf1_44 | 00000000 _H | 32 | FFD212CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 45 | RSCAN0TMD45 | 00000000 _H | 32 | FFD212D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 45 | RSCAN0TMPTR45 | 00000000 _H | 32 | FFD212D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 45 | RSCAN0TMDf0_45 | 00000000 _H | 32 | FFD212D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 45 | RSCAN0TMDf1_45 | 00000000 _H | 32 | FFD212DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 46 | RSCAN0TMD46 | 00000000 _H | 32 | FFD212E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 46 | RSCAN0TMPTR46 | 00000000 _H | 32 | FFD212E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 46 | RSCAN0TMDf0_46 | 00000000 _H | 32 | FFD212E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 46 | RSCAN0TMDf1_46 | 00000000 _H | 32 | FFD212EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer ID register 47 | RSCAN0TMD47 | 00000000 _H | 32 | FFD212F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer pointer register 47 | RSCAN0TMPTR47 | 00000000 _H | 32 | FFD212F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 0 register 47 | RSCAN0TMDf0_47 | 00000000 _H | 32 | FFD212F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit buffer data field 1 register 47 | RSCAN0TMDf1_47 | 00000000 _H | 32 | FFD212FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history access register 0 | RSCAN0THLACC0 | 00000000 _H | 32 | FFD21800 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history access register 1 | RSCAN0THLACC1 | 00000000 _H | 32 | FFD21804 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | Transmit history access register 2 | RSCAN0THLACC2 | 00000000 _H | 32 | FFD21808 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 0 | RSCAN0RPGACC0 | 00000000 _H | 32 | FFD21900 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 1 | RSCAN0RPGACC1 | 00000000 _H | 32 | FFD21904 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 2 | RSCAN0RPGACC2 | 00000000 _H | 32 | FFD21908 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 3 | RSCAN0RPGACC3 | 00000000 _H | 32 | FFD2190C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 4 | RSCAN0RPGACC4 | 00000000 _H | 32 | FFD21910 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 5 | RSCAN0RPGACC5 | 00000000 _H | 32 | FFD21914 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 6 | RSCAN0RPGACC6 | 00000000 _H | 32 | FFD21918 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 7 | RSCAN0RPGACC7 | 00000000 _H | 32 | FFD2191C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 8 | RSCAN0RPGACC8 | 00000000 _H | 32 | FFD21920 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 9 | RSCAN0RPGACC9 | 00000000 _H | 32 | FFD21924 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 10 | RSCAN0RPGACC10 | 00000000 _H | 32 | FFD21928 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (43/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|----------------------------------|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | RAM test page access register 11 | RSCAN0RPGACC11 | 00000000 _H | 32 | FFD2192C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 12 | RSCAN0RPGACC12 | 00000000 _H | 32 | FFD21930 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 13 | RSCAN0RPGACC13 | 00000000 _H | 32 | FFD21934 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 14 | RSCAN0RPGACC14 | 00000000 _H | 32 | FFD21938 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 15 | RSCAN0RPGACC15 | 00000000 _H | 32 | FFD2193C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 16 | RSCAN0RPGACC16 | 00000000 _H | 32 | FFD21940 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 17 | RSCAN0RPGACC17 | 00000000 _H | 32 | FFD21944 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 18 | RSCAN0RPGACC18 | 00000000 _H | 32 | FFD21948 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 19 | RSCAN0RPGACC19 | 00000000 _H | 32 | FFD2194C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 20 | RSCAN0RPGACC20 | 00000000 _H | 32 | FFD21950 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 21 | RSCAN0RPGACC21 | 00000000 _H | 32 | FFD21954 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 22 | RSCAN0RPGACC22 | 00000000 _H | 32 | FFD21958 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 23 | RSCAN0RPGACC23 | 00000000 _H | 32 | FFD2195C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 24 | RSCAN0RPGACC24 | 00000000 _H | 32 | FFD21960 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 25 | RSCAN0RPGACC25 | 00000000 _H | 32 | FFD21964 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 26 | RSCAN0RPGACC26 | 00000000 _H | 32 | FFD21968 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 27 | RSCAN0RPGACC27 | 00000000 _H | 32 | FFD2196C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 28 | RSCAN0RPGACC28 | 00000000 _H | 32 | FFD21970 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 29 | RSCAN0RPGACC29 | 00000000 _H | 32 | FFD21974 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 30 | RSCAN0RPGACC30 | 00000000 _H | 32 | FFD21978 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 31 | RSCAN0RPGACC31 | 00000000 _H | 32 | FFD2197C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 32 | RSCAN0RPGACC32 | 00000000 _H | 32 | FFD21980 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 33 | RSCAN0RPGACC33 | 00000000 _H | 32 | FFD21984 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 34 | RSCAN0RPGACC34 | 00000000 _H | 32 | FFD21988 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 35 | RSCAN0RPGACC35 | 00000000 _H | 32 | FFD2198C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 36 | RSCAN0RPGACC36 | 00000000 _H | 32 | FFD21990 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 37 | RSCAN0RPGACC37 | 00000000 _H | 32 | FFD21994 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 38 | RSCAN0RPGACC38 | 00000000 _H | 32 | FFD21998 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 39 | RSCAN0RPGACC39 | 00000000 _H | 32 | FFD2199C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 40 | RSCAN0RPGACC40 | 00000000 _H | 32 | FFD219A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 41 | RSCAN0RPGACC41 | 00000000 _H | 32 | FFD219A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 42 | RSCAN0RPGACC42 | 00000000 _H | 32 | FFD219A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 43 | RSCAN0RPGACC43 | 00000000 _H | 32 | FFD219AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 44 | RSCAN0RPGACC44 | 00000000 _H | 32 | FFD219B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 45 | RSCAN0RPGACC45 | 00000000 _H | 32 | FFD219B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 46 | RSCAN0RPGACC46 | 00000000 _H | 32 | FFD219B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 47 | RSCAN0RPGACC47 | 00000000 _H | 32 | FFD219BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 48 | RSCAN0RPGACC48 | 00000000 _H | 32 | FFD219C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 49 | RSCAN0RPGACC49 | 00000000 _H | 32 | FFD219C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 50 | RSCAN0RPGACC50 | 00000000 _H | 32 | FFD219C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 51 | RSCAN0RPGACC51 | 00000000 _H | 32 | FFD219CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 52 | RSCAN0RPGACC52 | 00000000 _H | 32 | FFD219D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 53 | RSCAN0RPGACC53 | 00000000 _H | 32 | FFD219D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 54 | RSCAN0RPGACC54 | 00000000 _H | 32 | FFD219D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 55 | RSCAN0RPGACC55 | 00000000 _H | 32 | FFD219DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 56 | RSCAN0RPGACC56 | 00000000 _H | 32 | FFD219E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 57 | RSCAN0RPGACC57 | 00000000 _H | 32 | FFD219E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 58 | RSCAN0RPGACC58 | 00000000 _H | 32 | FFD219E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (44/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCAN0 | RAM test page access register 59 | RSCAN0RPGACC59 | 00000000 _H | 32 | FFD219EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 60 | RSCAN0RPGACC60 | 00000000 _H | 32 | FFD219F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 61 | RSCAN0RPGACC61 | 00000000 _H | 32 | FFD219F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 62 | RSCAN0RPGACC62 | 00000000 _H | 32 | FFD219F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | RAM test page access register 63 | RSCAN0RPGACC63 | 00000000 _H | 32 | FFD219FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 0 Nominal Bitrate Configuration Register | RSCFD0CFDC0NCFG | 00000000 _H | 32 | FFD20000 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel control register | RSCFD0CFDC0CTR | 00000005 _H | 32 | FFD20004 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel status register | RSCFD0CFDC0STS | 00000005 _H | 32 | FFD20008 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel error flag register | RSCFD0CFDC0ERFL | 00000000 _H | 32 | FFD2000C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 1 Nominal Bitrate Configuration Register | RSCFD0CFDC1NCFG | 00000000 _H | 32 | FFD20010 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel control register | RSCFD0CFDC1CTR | 00000005 _H | 32 | FFD20014 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel status register | RSCFD0CFDC1STS | 00000005 _H | 32 | FFD20018 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel error flag register | RSCFD0CFDC1ERFL | 00000000 _H | 32 | FFD2001C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 2 Nominal Bitrate Configuration Register | RSCFD0CFDC2NCFG | 00000000 _H | 32 | FFD20020 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel control register | RSCFD0CFDC2CTR | 00000005 _H | 32 | FFD20024 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel status register | RSCFD0CFDC2STS | 00000005 _H | 32 | FFD20028 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel error flag register | RSCFD0CFDC2ERFL | 00000000 _H | 32 | FFD2002C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global configuration register | RSCFD0CFDGCFCG | 00000000 _H | 32 | FFD20084 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global control register | RSCFD0CFDGCCTR | 00000005 _H | 32 | FFD20088 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global status register | RSCFD0CFDGCSTS | 0000000D _H | 32 | FFD2008C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global error flag register | RSCFD0CFDGERFL | 00000000 _H | 32 | FFD20090 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global timestamp counter register | RSCFD0CFDGTSC | 00000000 _H | 32 | FFD20094 _H | 2 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive rule entry control register | RSCFD0CFDGALECTR | 00000000 _H | 32 | FFD20098 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive rule configuration register 0 | RSCFD0CFDGAFLCFG0 | 00000000 _H | 32 | FFD2009C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer number register | RSCFD0CFDRMNB | 00000000 _H | 32 | FFD200A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer new data register 0 | RSCFD0CFDRMND0 | 00000000 _H | 32 | FFD200A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer new data register 1 | RSCFD0CFDRMND1 | 00000000 _H | 32 | FFD200AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 0 | RSCFD0CFDRFCC0 | 00000000 _H | 32 | FFD200B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 1 | RSCFD0CFDRFCC1 | 00000000 _H | 32 | FFD200BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 2 | RSCFD0CFDRFCC2 | 00000000 _H | 32 | FFD200C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 3 | RSCFD0CFDRFCC3 | 00000000 _H | 32 | FFD200C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 4 | RSCFD0CFDRFCC4 | 00000000 _H | 32 | FFD200C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 5 | RSCFD0CFDRFCC5 | 00000000 _H | 32 | FFD200CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 6 | RSCFD0CFDRFCC6 | 00000000 _H | 32 | FFD200D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer configuration and control register 7 | RSCFD0CFDRFCC7 | 00000000 _H | 32 | FFD200D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 0 | RSCFD0CFDRFSTS0 | 00000001 _H | 32 | FFD200D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 1 | RSCFD0CFDRFSTS1 | 00000001 _H | 32 | FFD200DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 2 | RSCFD0CFDRFSTS2 | 00000001 _H | 32 | FFD200E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 3 | RSCFD0CFDRFSTS3 | 00000001 _H | 32 | FFD200E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 4 | RSCFD0CFDRFSTS4 | 00000001 _H | 32 | FFD200E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 5 | RSCFD0CFDRFSTS5 | 00000001 _H | 32 | FFD200EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 6 | RSCFD0CFDRFSTS6 | 00000001 _H | 32 | FFD200F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer status register 7 | RSCFD0CFDRFSTS7 | 00000001 _H | 32 | FFD200F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer pointer control register 0 | RSCFD0CFDRFPCTR0 | 00000000 _H | 32 | FFD200F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer pointer control register 1 | RSCFD0CFDRFPCTR1 | 00000000 _H | 32 | FFD200FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (45/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Receive FIFO buffer pointer control register 2 | RSCFD0CFDRFPCTR2 | 00000000 _H | 32 | FFD20100 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer pointer control register 3 | RSCFD0CFDRFPCTR3 | 00000000 _H | 32 | FFD20104 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer pointer control register 4 | RSCFD0CFDRFPCTR4 | 00000000 _H | 32 | FFD20108 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer pointer control register 5 | RSCFD0CFDRFPCTR5 | 00000000 _H | 32 | FFD2010C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer pointer control register 6 | RSCFD0CFDRFPCTR6 | 00000000 _H | 32 | FFD20110 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer pointer control register 7 | RSCFD0CFDRFPCTR7 | 00000000 _H | 32 | FFD20114 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 0 | RSCFD0CFDCFCFC0 | 00000000 _H | 32 | FFD20118 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 1 | RSCFD0CFDCFCFC1 | 00000000 _H | 32 | FFD2011C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 2 | RSCFD0CFDCFCFC2 | 00000000 _H | 32 | FFD20120 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 3 | RSCFD0CFDCFCFC3 | 00000000 _H | 32 | FFD20124 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 4 | RSCFD0CFDCFCFC4 | 00000000 _H | 32 | FFD20128 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 5 | RSCFD0CFDCFCFC5 | 00000000 _H | 32 | FFD2012C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 6 | RSCFD0CFDCFCFC6 | 00000000 _H | 32 | FFD20130 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 7 | RSCFD0CFDCFCFC7 | 00000000 _H | 32 | FFD20134 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer configuration and control register 8 | RSCFD0CFDCFCFC8 | 00000000 _H | 32 | FFD20138 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 0 | RSCFD0CFDCFCSTS0 | 00000001 _H | 32 | FFD20178 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 1 | RSCFD0CFDCFCSTS1 | 00000001 _H | 32 | FFD2017C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 2 | RSCFD0CFDCFCSTS2 | 00000001 _H | 32 | FFD20180 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 3 | RSCFD0CFDCFCSTS3 | 00000001 _H | 32 | FFD20184 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 4 | RSCFD0CFDCFCSTS4 | 00000001 _H | 32 | FFD20188 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 5 | RSCFD0CFDCFCSTS5 | 00000001 _H | 32 | FFD2018C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 6 | RSCFD0CFDCFCSTS6 | 00000001 _H | 32 | FFD20190 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 7 | RSCFD0CFDCFCSTS7 | 00000001 _H | 32 | FFD20194 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer status register 8 | RSCFD0CFDCFCSTS8 | 00000001 _H | 32 | FFD20198 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 0 | RSCFD0CFDCFPCTR0 | 00000000 _H | 32 | FFD201D8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 1 | RSCFD0CFDCFPCTR1 | 00000000 _H | 32 | FFD201DC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 2 | RSCFD0CFDCFPCTR2 | 00000000 _H | 32 | FFD201E0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 3 | RSCFD0CFDCFPCTR3 | 00000000 _H | 32 | FFD201E4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 4 | RSCFD0CFDCFPCTR4 | 00000000 _H | 32 | FFD201E8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 5 | RSCFD0CFDCFPCTR5 | 00000000 _H | 32 | FFD201EC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 6 | RSCFD0CFDCFPCTR6 | 00000000 _H | 32 | FFD201F0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 7 | RSCFD0CFDCFPCTR7 | 00000000 _H | 32 | FFD201F4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer pointer control register 8 | RSCFD0CFDCFPCTR8 | 00000000 _H | 32 | FFD201F8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | FIFO empty status register | RSCFD0CFDFESTS | 03FFFFFF _H | 32 | FFD20238 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | FIFO full status register | RSCFD0CFDFESTS | 00000000 _H | 32 | FFD2023C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | FIFO message lost status register | RSCFD0CFDFMSTS | 00000000 _H | 32 | FFD20240 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer interrupt flag status register | RSCFD0CFDRFISTS | 00000000 _H | 32 | FFD20244 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer RX interrupt flag status register | RSCFD0CFDCFRISTS | 00000000 _H | 32 | FFD20248 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer TX interrupt flag status register | RSCFD0CFDCFTISTS | 00000000 _H | 32 | FFD2024C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit buffer control register 0 | RSCFD0CFDTMTC0 | 00 _H | 8 | FFD20250 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit buffer control register 1 | RSCFD0CFDTMTC1 | 00 _H | 8 | FFD20254 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (46/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|-------------------------------------|-----------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer control register 2 | RSCFD0CFDTC2 | 00 _H | 8 | FFD20252 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 3 | RSCFD0CFDTC3 | 00 _H | 8 | FFD20253 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 4 | RSCFD0CFDTC4 | 00 _H | 8 | FFD20254 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 5 | RSCFD0CFDTC5 | 00 _H | 8 | FFD20255 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 6 | RSCFD0CFDTC6 | 00 _H | 8 | FFD20256 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 7 | RSCFD0CFDTC7 | 00 _H | 8 | FFD20257 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 8 | RSCFD0CFDTC8 | 00 _H | 8 | FFD20258 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 9 | RSCFD0CFDTC9 | 00 _H | 8 | FFD20259 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 10 | RSCFD0CFDTC10 | 00 _H | 8 | FFD2025A _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 11 | RSCFD0CFDTC11 | 00 _H | 8 | FFD2025B _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 12 | RSCFD0CFDTC12 | 00 _H | 8 | FFD2025C _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 13 | RSCFD0CFDTC13 | 00 _H | 8 | FFD2025D _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 14 | RSCFD0CFDTC14 | 00 _H | 8 | FFD2025E _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 15 | RSCFD0CFDTC15 | 00 _H | 8 | FFD2025F _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 16 | RSCFD0CFDTC16 | 00 _H | 8 | FFD20260 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 17 | RSCFD0CFDTC17 | 00 _H | 8 | FFD20261 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 18 | RSCFD0CFDTC18 | 00 _H | 8 | FFD20262 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 19 | RSCFD0CFDTC19 | 00 _H | 8 | FFD20263 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 20 | RSCFD0CFDTC20 | 00 _H | 8 | FFD20264 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 21 | RSCFD0CFDTC21 | 00 _H | 8 | FFD20265 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 22 | RSCFD0CFDTC22 | 00 _H | 8 | FFD20266 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 23 | RSCFD0CFDTC23 | 00 _H | 8 | FFD20267 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 24 | RSCFD0CFDTC24 | 00 _H | 8 | FFD20268 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 25 | RSCFD0CFDTC25 | 00 _H | 8 | FFD20269 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 26 | RSCFD0CFDTC26 | 00 _H | 8 | FFD2026A _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 27 | RSCFD0CFDTC27 | 00 _H | 8 | FFD2026B _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 28 | RSCFD0CFDTC28 | 00 _H | 8 | FFD2026C _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 29 | RSCFD0CFDTC29 | 00 _H | 8 | FFD2026D _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 30 | RSCFD0CFDTC30 | 00 _H | 8 | FFD2026E _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 31 | RSCFD0CFDTC31 | 00 _H | 8 | FFD2026F _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 32 | RSCFD0CFDTC32 | 00 _H | 8 | FFD20270 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 33 | RSCFD0CFDTC33 | 00 _H | 8 | FFD20271 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 34 | RSCFD0CFDTC34 | 00 _H | 8 | FFD20272 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 35 | RSCFD0CFDTC35 | 00 _H | 8 | FFD20273 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 36 | RSCFD0CFDTC36 | 00 _H | 8 | FFD20274 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 37 | RSCFD0CFDTC37 | 00 _H | 8 | FFD20275 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 38 | RSCFD0CFDTC38 | 00 _H | 8 | FFD20276 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 39 | RSCFD0CFDTC39 | 00 _H | 8 | FFD20277 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 40 | RSCFD0CFDTC40 | 00 _H | 8 | FFD20278 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 41 | RSCFD0CFDTC41 | 00 _H | 8 | FFD20279 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 42 | RSCFD0CFDTC42 | 00 _H | 8 | FFD2027A _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 43 | RSCFD0CFDTC43 | 00 _H | 8 | FFD2027B _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 44 | RSCFD0CFDTC44 | 00 _H | 8 | FFD2027C _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 45 | RSCFD0CFDTC45 | 00 _H | 8 | FFD2027D _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 46 | RSCFD0CFDTC46 | 00 _H | 8 | FFD2027E _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer control register 47 | RSCFD0CFDTC47 | 00 _H | 8 | FFD2027F _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 0 | RSCFD0CFDTCMST0 | 00 _H | 8 | FFD202D0 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 1 | RSCFD0CFDTCMST1 | 00 _H | 8 | FFD202D1 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (47/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, ---: not applicable) | | | | Access Authority | |
|--------|--|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer status register 2 | RSCFD0CFDTMSTS2 | 00 _H | 8 | FFD202D2 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 3 | RSCFD0CFDTMSTS3 | 00 _H | 8 | FFD202D3 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 4 | RSCFD0CFDTMSTS4 | 00 _H | 8 | FFD202D4 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 5 | RSCFD0CFDTMSTS5 | 00 _H | 8 | FFD202D5 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 6 | RSCFD0CFDTMSTS6 | 00 _H | 8 | FFD202D6 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 7 | RSCFD0CFDTMSTS7 | 00 _H | 8 | FFD202D7 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 8 | RSCFD0CFDTMSTS8 | 00 _H | 8 | FFD202D8 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 9 | RSCFD0CFDTMSTS9 | 00 _H | 8 | FFD202D9 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 10 | RSCFD0CFDTMSTS10 | 00 _H | 8 | FFD202DA _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 11 | RSCFD0CFDTMSTS11 | 00 _H | 8 | FFD202DB _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 12 | RSCFD0CFDTMSTS12 | 00 _H | 8 | FFD202DC _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 13 | RSCFD0CFDTMSTS13 | 00 _H | 8 | FFD202DD _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 14 | RSCFD0CFDTMSTS14 | 00 _H | 8 | FFD202DE _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 15 | RSCFD0CFDTMSTS15 | 00 _H | 8 | FFD202DF _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 16 | RSCFD0CFDTMSTS16 | 00 _H | 8 | FFD202E0 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 17 | RSCFD0CFDTMSTS17 | 00 _H | 8 | FFD202E1 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 18 | RSCFD0CFDTMSTS18 | 00 _H | 8 | FFD202E2 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 19 | RSCFD0CFDTMSTS19 | 00 _H | 8 | FFD202E3 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 20 | RSCFD0CFDTMSTS20 | 00 _H | 8 | FFD202E4 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 21 | RSCFD0CFDTMSTS21 | 00 _H | 8 | FFD202E5 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 22 | RSCFD0CFDTMSTS22 | 00 _H | 8 | FFD202E6 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 23 | RSCFD0CFDTMSTS23 | 00 _H | 8 | FFD202E7 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 24 | RSCFD0CFDTMSTS24 | 00 _H | 8 | FFD202E8 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 25 | RSCFD0CFDTMSTS25 | 00 _H | 8 | FFD202E9 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 26 | RSCFD0CFDTMSTS26 | 00 _H | 8 | FFD202EA _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 27 | RSCFD0CFDTMSTS27 | 00 _H | 8 | FFD202EB _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 28 | RSCFD0CFDTMSTS28 | 00 _H | 8 | FFD202EC _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 29 | RSCFD0CFDTMSTS29 | 00 _H | 8 | FFD202ED _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 30 | RSCFD0CFDTMSTS30 | 00 _H | 8 | FFD202EE _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 31 | RSCFD0CFDTMSTS31 | 00 _H | 8 | FFD202EF _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 32 | RSCFD0CFDTMSTS32 | 00 _H | 8 | FFD202F0 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 33 | RSCFD0CFDTMSTS33 | 00 _H | 8 | FFD202F1 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 34 | RSCFD0CFDTMSTS34 | 00 _H | 8 | FFD202F2 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 35 | RSCFD0CFDTMSTS35 | 00 _H | 8 | FFD202F3 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 36 | RSCFD0CFDTMSTS36 | 00 _H | 8 | FFD202F4 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 37 | RSCFD0CFDTMSTS37 | 00 _H | 8 | FFD202F5 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 38 | RSCFD0CFDTMSTS38 | 00 _H | 8 | FFD202F6 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 39 | RSCFD0CFDTMSTS39 | 00 _H | 8 | FFD202F7 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 40 | RSCFD0CFDTMSTS40 | 00 _H | 8 | FFD202F8 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 41 | RSCFD0CFDTMSTS41 | 00 _H | 8 | FFD202F9 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 42 | RSCFD0CFDTMSTS42 | 00 _H | 8 | FFD202FA _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 43 | RSCFD0CFDTMSTS43 | 00 _H | 8 | FFD202FB _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 44 | RSCFD0CFDTMSTS44 | 00 _H | 8 | FFD202FC _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 45 | RSCFD0CFDTMSTS45 | 00 _H | 8 | FFD202FD _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 46 | RSCFD0CFDTMSTS46 | 00 _H | 8 | FFD202FE _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer status register 47 | RSCFD0CFDTMSTS47 | 00 _H | 8 | FFD202FF _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer transmit request status register 0 | RSCFD0CFDTMTRSTS0 | 00000000 _H | 32 | FFD20350 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer transmit request status register 1 | RSCFD0CFDTMTRSTS1 | 00000000 _H | 32 | FFD20354 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (48/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer transmit abort request status register 0 | RSCFD0CFDXTMTARSTS0 | 00000000 _H | 32 | FFD20360 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer transmit abort request status register 1 | RSCFD0CFDXTMTARSTS1 | 00000000 _H | 32 | FFD20364 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer transmit complete status register 0 | RSCFD0CFDXTMTCSTS0 | 00000000 _H | 32 | FFD20370 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer transmit complete status register 1 | RSCFD0CFDXTMTCSTS1 | 00000000 _H | 32 | FFD20374 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer transmit abort status register 0 | RSCFD0CFDXTMTASTS0 | 00000000 _H | 32 | FFD20380 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer transmit abort status register 1 | RSCFD0CFDXTMTASTS1 | 00000000 _H | 32 | FFD20384 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer interrupt enable configuration register 0 | RSCFD0CFDXTMIEC0 | 00000000 _H | 32 | FFD20390 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer interrupt enable configuration register 1 | RSCFD0CFDXTMIEC1 | 00000000 _H | 32 | FFD20394 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue configuration and control register 0 | RSCFD0CFDXTMQCC0 | 00000000 _H | 32 | FFD203A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue configuration and control register 1 | RSCFD0CFDXTMQCC1 | 00000000 _H | 32 | FFD203A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue configuration and control register 2 | RSCFD0CFDXTMQCC2 | 00000000 _H | 32 | FFD203A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue status register 0 | RSCFD0CFDXTMQSTS0 | 00000001 _H | 32 | FFD203C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue status register 1 | RSCFD0CFDXTMQSTS1 | 00000001 _H | 32 | FFD203C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue status register 2 | RSCFD0CFDXTMQSTS2 | 00000001 _H | 32 | FFD203C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue pointer control register 0 | RSCFD0CFDXTQPCTR0 | 00000000 _H | 32 | FFD203E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue pointer control register 1 | RSCFD0CFDXTQPCTR1 | 00000000 _H | 32 | FFD203E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit queue pointer control register 2 | RSCFD0CFDXTQPCTR2 | 00000000 _H | 32 | FFD203E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history configuration and control register 0 | RSCFD0CFDXTMLCC0 | 00000000 _H | 32 | FFD20400 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history configuration and control register 1 | RSCFD0CFDXTMLCC1 | 00000000 _H | 32 | FFD20404 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history configuration and control register 2 | RSCFD0CFDXTMLCC2 | 00000000 _H | 32 | FFD20408 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history status register 0 | RSCFD0CFDXTMLSTS0 | 00000001 _H | 32 | FFD20420 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history status register 1 | RSCFD0CFDXTMLSTS1 | 00000001 _H | 32 | FFD20424 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history status register 2 | RSCFD0CFDXTMLSTS2 | 00000001 _H | 32 | FFD20428 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history pointer control register 0 | RSCFD0CFDXTMLPCTR0 | 00000000 _H | 32 | FFD20440 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history pointer control register 1 | RSCFD0CFDXTMLPCTR1 | 00000000 _H | 32 | FFD20444 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history pointer control register 2 | RSCFD0CFDXTMLPCTR2 | 00000000 _H | 32 | FFD20448 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global TX interrupt status register 0 | RSCFD0CFDXTINTSTS0 | 00000000 _H | 32 | FFD20460 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global test configuration register | RSCFD0CFDXTSTCFG | 00000000 _H | 32 | FFD20468 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global test control register | RSCFD0CFDXTSTCTR | 00000000 _H | 32 | FFD2046C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global FD Configuration register | RSCFD0CFDGFDCFG | 00000000 _H | 32 | FFD20474 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global lock key register | RSCFD0CFDGLCKK | 00000000 _H | 32 | FFD2047C _H | 2 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | DMA Transfer Control Register | RSCFD0CFDCTCT | 00000000 _H | 32 | FFD20490 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | DMA Transfer Status Register | RSCFD0CFDCTSTS | 00000000 _H | 32 | FFD20494 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Global Register Map Configuration Register | RSCFD0CFDGRMCFG | 00000000 _H | 32 | FFD204FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 0 Data Bitrate Configuration Register | RSCFD0CFDC0DCFG | 00000000 _H | 32 | FFD20500 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 0 CAN FD Configuration Register | RSCFD0CFDC0FCFG | 00000000 _H | 32 | FFD20504 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 0 CAN FD Control Register | RSCFD0CFDC0FDCTR | 00000000 _H | 32 | FFD20508 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 0 CAN FD Status Register | RSCFD0CFDC0FDSTS | 00000000 _H | 32 | FFD2050C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 0 CAN FD CRC Register | RSCFD0CFDC0FDCRC | 00000000 _H | 32 | FFD20510 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 1 Data Bitrate Configuration Register | RSCFD0CFDC1DCFG | 00000000 _H | 32 | FFD20520 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 1 CAN FD Configuration Register | RSCFD0CFDC1FCFG | 00000000 _H | 32 | FFD20524 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 1 CAN FD Control Register | RSCFD0CFDC1FDCTR | 00000000 _H | 32 | FFD20528 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 1 CAN FD Status Register | RSCFD0CFDC1FDSTS | 00000000 _H | 32 | FFD2052C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Channel 1 CAN FD CRC Register | RSCFD0CFDC1FDCRC | 00000000 _H | 32 | FFD20530 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (49/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Channel 2 Data Bitrate Configuration Register | RSCFD0CFDC2DCFG | 00000000 _H | 32 | FFD20540 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Channel 2 CAN FD Configuration Register | RSCFD0CFDC2FDCFG | 00000000 _H | 32 | FFD20544 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Channel 2 CAN FD Control Register | RSCFD0CFDC2FDCTR | 00000000 _H | 32 | FFD20548 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Channel 2 CAN FD Status Register | RSCFD0CFDC2FDSTS | 00000000 _H | 32 | FFD2054C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Channel 2 CAN FD CRC Register | RSCFD0CFDC2FDCRC | 00000000 _H | 32 | FFD20550 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 0 | RSCFD0CFDGAFLID0 | 00000000 _H | 32 | FFD21000 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 0 | RSCFD0CFDGAFLM0 | 00000000 _H | 32 | FFD21004 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 0 | RSCFD0CFDGAFLP0_0 | 00000000 _H | 32 | FFD21008 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 0 | RSCFD0CFDGAFLP1_0 | 00000000 _H | 32 | FFD2100C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 1 | RSCFD0CFDGAFLID1 | 00000000 _H | 32 | FFD21010 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 1 | RSCFD0CFDGAFLM1 | 00000000 _H | 32 | FFD21014 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 1 | RSCFD0CFDGAFLP0_1 | 00000000 _H | 32 | FFD21018 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 1 | RSCFD0CFDGAFLP1_1 | 00000000 _H | 32 | FFD2101C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 2 | RSCFD0CFDGAFLID2 | 00000000 _H | 32 | FFD21020 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 2 | RSCFD0CFDGAFLM2 | 00000000 _H | 32 | FFD21024 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 2 | RSCFD0CFDGAFLP0_2 | 00000000 _H | 32 | FFD21028 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 2 | RSCFD0CFDGAFLP1_2 | 00000000 _H | 32 | FFD2102C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 3 | RSCFD0CFDGAFLID3 | 00000000 _H | 32 | FFD21030 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 3 | RSCFD0CFDGAFLM3 | 00000000 _H | 32 | FFD21034 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 3 | RSCFD0CFDGAFLP0_3 | 00000000 _H | 32 | FFD21038 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 3 | RSCFD0CFDGAFLP1_3 | 00000000 _H | 32 | FFD2103C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 4 | RSCFD0CFDGAFLID4 | 00000000 _H | 32 | FFD21040 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 4 | RSCFD0CFDGAFLM4 | 00000000 _H | 32 | FFD21044 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 4 | RSCFD0CFDGAFLP0_4 | 00000000 _H | 32 | FFD21048 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 4 | RSCFD0CFDGAFLP1_4 | 00000000 _H | 32 | FFD2104C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 5 | RSCFD0CFDGAFLID5 | 00000000 _H | 32 | FFD21050 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 5 | RSCFD0CFDGAFLM5 | 00000000 _H | 32 | FFD21054 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 5 | RSCFD0CFDGAFLP0_5 | 00000000 _H | 32 | FFD21058 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 5 | RSCFD0CFDGAFLP1_5 | 00000000 _H | 32 | FFD2105C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 6 | RSCFD0CFDGAFLID6 | 00000000 _H | 32 | FFD21060 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 6 | RSCFD0CFDGAFLM6 | 00000000 _H | 32 | FFD21064 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 6 | RSCFD0CFDGAFLP0_6 | 00000000 _H | 32 | FFD21068 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 6 | RSCFD0CFDGAFLP1_6 | 00000000 _H | 32 | FFD2106C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 7 | RSCFD0CFDGAFLID7 | 00000000 _H | 32 | FFD21070 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 7 | RSCFD0CFDGAFLM7 | 00000000 _H | 32 | FFD21074 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 7 | RSCFD0CFDGAFLP0_7 | 00000000 _H | 32 | FFD21078 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 7 | RSCFD0CFDGAFLP1_7 | 00000000 _H | 32 | FFD2107C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 8 | RSCFD0CFDGAFLID8 | 00000000 _H | 32 | FFD21080 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 8 | RSCFD0CFDGAFLM8 | 00000000 _H | 32 | FFD21084 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 8 | RSCFD0CFDGAFLP0_8 | 00000000 _H | 32 | FFD21088 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 8 | RSCFD0CFDGAFLP1_8 | 00000000 _H | 32 | FFD2108C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 9 | RSCFD0CFDGAFLID9 | 00000000 _H | 32 | FFD21090 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 9 | RSCFD0CFDGAFLM9 | 00000000 _H | 32 | FFD21094 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 9 | RSCFD0CFDGAFLP0_9 | 00000000 _H | 32 | FFD21098 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 1 Register 9 | RSCFD0CFDGAFLP1_9 | 00000000 _H | 32 | FFD2109C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule ID register 10 | RSCFD0CFDGAFLID10 | 00000000 _H | 32 | FFD210A0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive rule mask register 10 | RSCFD0CFDGAFLM10 | 00000000 _H | 32 | FFD210A4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Global Acceptance Filter List Pointer 0 Register 10 | RSCFD0CFDGAFLP0_10 | 00000000 _H | 32 | FFD210A8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (55/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | RX Message Buffer Data Field 0 Register 27 | RSCFD0CFDRMDF0_27 | 00000000 _H | 32 | FFD2236C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 27 | RSCFD0CFDRMDF1_27 | 00000000 _H | 32 | FFD22370 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 27 | RSCFD0CFDRMDF2_27 | 00000000 _H | 32 | FFD22374 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 27 | RSCFD0CFDRMDF3_27 | 00000000 _H | 32 | FFD22378 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 27 | RSCFD0CFDRMDF4_27 | 00000000 _H | 32 | FFD2237C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 28 | RSCFD0CFDRMID28 | 00000000 _H | 32 | FFD22380 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 28 | RSCFD0CFDRMPTR28 | 00000000 _H | 32 | FFD22384 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 28 | RSCFD0CFDRMF DSTS28 | 00000000 _H | 32 | FFD22388 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 28 | RSCFD0CFDRMDF0_28 | 00000000 _H | 32 | FFD2238C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 28 | RSCFD0CFDRMDF1_28 | 00000000 _H | 32 | FFD22390 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 28 | RSCFD0CFDRMDF2_28 | 00000000 _H | 32 | FFD22394 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 28 | RSCFD0CFDRMDF3_28 | 00000000 _H | 32 | FFD22398 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 28 | RSCFD0CFDRMDF4_28 | 00000000 _H | 32 | FFD2239C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 29 | RSCFD0CFDRMID29 | 00000000 _H | 32 | FFD223A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 29 | RSCFD0CFDRMPTR29 | 00000000 _H | 32 | FFD223A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 29 | RSCFD0CFDRMF DSTS29 | 00000000 _H | 32 | FFD223A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 29 | RSCFD0CFDRMDF0_29 | 00000000 _H | 32 | FFD223AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 29 | RSCFD0CFDRMDF1_29 | 00000000 _H | 32 | FFD223B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 29 | RSCFD0CFDRMDF2_29 | 00000000 _H | 32 | FFD223B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 29 | RSCFD0CFDRMDF3_29 | 00000000 _H | 32 | FFD223B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 29 | RSCFD0CFDRMDF4_29 | 00000000 _H | 32 | FFD223BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 30 | RSCFD0CFDRMID30 | 00000000 _H | 32 | FFD223C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 30 | RSCFD0CFDRMPTR30 | 00000000 _H | 32 | FFD223C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 30 | RSCFD0CFDRMF DSTS30 | 00000000 _H | 32 | FFD223C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 30 | RSCFD0CFDRMDF0_30 | 00000000 _H | 32 | FFD223CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 30 | RSCFD0CFDRMDF1_30 | 00000000 _H | 32 | FFD223D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 30 | RSCFD0CFDRMDF2_30 | 00000000 _H | 32 | FFD223D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 30 | RSCFD0CFDRMDF3_30 | 00000000 _H | 32 | FFD223D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 30 | RSCFD0CFDRMDF4_30 | 00000000 _H | 32 | FFD223DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 31 | RSCFD0CFDRMID31 | 00000000 _H | 32 | FFD223E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 31 | RSCFD0CFDRMPTR31 | 00000000 _H | 32 | FFD223E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 31 | RSCFD0CFDRMF DSTS31 | 00000000 _H | 32 | FFD223E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 31 | RSCFD0CFDRMDF0_31 | 00000000 _H | 32 | FFD223EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 31 | RSCFD0CFDRMDF1_31 | 00000000 _H | 32 | FFD223F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 31 | RSCFD0CFDRMDF2_31 | 00000000 _H | 32 | FFD223F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 31 | RSCFD0CFDRMDF3_31 | 00000000 _H | 32 | FFD223F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 31 | RSCFD0CFDRMDF4_31 | 00000000 _H | 32 | FFD223FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 32 | RSCFD0CFDRMID32 | 00000000 _H | 32 | FFD22400 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 32 | RSCFD0CFDRMPTR32 | 00000000 _H | 32 | FFD22404 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 32 | RSCFD0CFDRMF DSTS32 | 00000000 _H | 32 | FFD22408 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 32 | RSCFD0CFDRMDF0_32 | 00000000 _H | 32 | FFD2240C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 32 | RSCFD0CFDRMDF1_32 | 00000000 _H | 32 | FFD22410 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 32 | RSCFD0CFDRMDF2_32 | 00000000 _H | 32 | FFD22414 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 32 | RSCFD0CFDRMDF3_32 | 00000000 _H | 32 | FFD22418 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 32 | RSCFD0CFDRMDF4_32 | 00000000 _H | 32 | FFD2241C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 33 | RSCFD0CFDRMID33 | 00000000 _H | 32 | FFD22420 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 33 | RSCFD0CFDRMPTR33 | 00000000 _H | 32 | FFD22424 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 33 | RSCFD0CFDRMF DSTS33 | 00000000 _H | 32 | FFD22428 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (56/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | RX Message Buffer Data Field 0 Register 33 | RSCFD0CFDRMDF0_33 | 00000000 _H | 32 | FFD2242C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 33 | RSCFD0CFDRMDF1_33 | 00000000 _H | 32 | FFD22430 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 33 | RSCFD0CFDRMDF2_33 | 00000000 _H | 32 | FFD22434 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 33 | RSCFD0CFDRMDF3_33 | 00000000 _H | 32 | FFD22438 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 33 | RSCFD0CFDRMDF4_33 | 00000000 _H | 32 | FFD2243C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer ID register 34 | RSCFD0CFDRMID34 | 00000000 _H | 32 | FFD22440 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer pointer register 34 | RSCFD0CFDRMPTR34 | 00000000 _H | 32 | FFD22444 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 34 | RSCFD0CFDRMFSDSTS34 | 00000000 _H | 32 | FFD22448 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 34 | RSCFD0CFDRMDF0_34 | 00000000 _H | 32 | FFD2244C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 34 | RSCFD0CFDRMDF1_34 | 00000000 _H | 32 | FFD22450 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 34 | RSCFD0CFDRMDF2_34 | 00000000 _H | 32 | FFD22454 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 34 | RSCFD0CFDRMDF3_34 | 00000000 _H | 32 | FFD22458 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 34 | RSCFD0CFDRMDF4_34 | 00000000 _H | 32 | FFD2245C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer ID register 35 | RSCFD0CFDRMID35 | 00000000 _H | 32 | FFD22460 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer pointer register 35 | RSCFD0CFDRMPTR35 | 00000000 _H | 32 | FFD22464 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 35 | RSCFD0CFDRMFSDSTS35 | 00000000 _H | 32 | FFD22468 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 35 | RSCFD0CFDRMDF0_35 | 00000000 _H | 32 | FFD2246C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 35 | RSCFD0CFDRMDF1_35 | 00000000 _H | 32 | FFD22470 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 35 | RSCFD0CFDRMDF2_35 | 00000000 _H | 32 | FFD22474 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 35 | RSCFD0CFDRMDF3_35 | 00000000 _H | 32 | FFD22478 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 35 | RSCFD0CFDRMDF4_35 | 00000000 _H | 32 | FFD2247C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer ID register 36 | RSCFD0CFDRMID36 | 00000000 _H | 32 | FFD22480 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer pointer register 36 | RSCFD0CFDRMPTR36 | 00000000 _H | 32 | FFD22484 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 36 | RSCFD0CFDRMFSDSTS36 | 00000000 _H | 32 | FFD22488 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 36 | RSCFD0CFDRMDF0_36 | 00000000 _H | 32 | FFD2248C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 36 | RSCFD0CFDRMDF1_36 | 00000000 _H | 32 | FFD22490 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 36 | RSCFD0CFDRMDF2_36 | 00000000 _H | 32 | FFD22494 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 36 | RSCFD0CFDRMDF3_36 | 00000000 _H | 32 | FFD22498 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 36 | RSCFD0CFDRMDF4_36 | 00000000 _H | 32 | FFD2249C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer ID register 37 | RSCFD0CFDRMID37 | 00000000 _H | 32 | FFD224A0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer pointer register 37 | RSCFD0CFDRMPTR37 | 00000000 _H | 32 | FFD224A4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 37 | RSCFD0CFDRMFSDSTS37 | 00000000 _H | 32 | FFD224A8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 37 | RSCFD0CFDRMDF0_37 | 00000000 _H | 32 | FFD224AC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 37 | RSCFD0CFDRMDF1_37 | 00000000 _H | 32 | FFD224B0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 37 | RSCFD0CFDRMDF2_37 | 00000000 _H | 32 | FFD224B4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 37 | RSCFD0CFDRMDF3_37 | 00000000 _H | 32 | FFD224B8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 37 | RSCFD0CFDRMDF4_37 | 00000000 _H | 32 | FFD224BC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer ID register 38 | RSCFD0CFDRMID38 | 00000000 _H | 32 | FFD224C0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer pointer register 38 | RSCFD0CFDRMPTR38 | 00000000 _H | 32 | FFD224C4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 38 | RSCFD0CFDRMFSDSTS38 | 00000000 _H | 32 | FFD224C8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 38 | RSCFD0CFDRMDF0_38 | 00000000 _H | 32 | FFD224CC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 38 | RSCFD0CFDRMDF1_38 | 00000000 _H | 32 | FFD224D0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 38 | RSCFD0CFDRMDF2_38 | 00000000 _H | 32 | FFD224D4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 38 | RSCFD0CFDRMDF3_38 | 00000000 _H | 32 | FFD224D8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 38 | RSCFD0CFDRMDF4_38 | 00000000 _H | 32 | FFD224DC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer ID register 39 | RSCFD0CFDRMID39 | 00000000 _H | 32 | FFD224E0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive buffer pointer register 39 | RSCFD0CFDRMPTR39 | 00000000 _H | 32 | FFD224E4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 39 | RSCFD0CFDRMFSDSTS39 | 00000000 _H | 32 | FFD224E8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (57/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | RX Message Buffer Data Field 0 Register 39 | RSCFD0CFDRMDF0_39 | 00000000 _H | 32 | FFD224EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 39 | RSCFD0CFDRMDF1_39 | 00000000 _H | 32 | FFD224F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 39 | RSCFD0CFDRMDF2_39 | 00000000 _H | 32 | FFD224F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 39 | RSCFD0CFDRMDF3_39 | 00000000 _H | 32 | FFD224F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 39 | RSCFD0CFDRMDF4_39 | 00000000 _H | 32 | FFD224FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 40 | RSCFD0CFDRMID40 | 00000000 _H | 32 | FFD22500 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 40 | RSCFD0CFDRMPTR40 | 00000000 _H | 32 | FFD22504 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 40 | RSCFD0CFDRMFST40 | 00000000 _H | 32 | FFD22508 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 40 | RSCFD0CFDRMDF0_40 | 00000000 _H | 32 | FFD2250C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 40 | RSCFD0CFDRMDF1_40 | 00000000 _H | 32 | FFD22510 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 40 | RSCFD0CFDRMDF2_40 | 00000000 _H | 32 | FFD22514 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 40 | RSCFD0CFDRMDF3_40 | 00000000 _H | 32 | FFD22518 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 40 | RSCFD0CFDRMDF4_40 | 00000000 _H | 32 | FFD2251C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 41 | RSCFD0CFDRMID41 | 00000000 _H | 32 | FFD22520 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 41 | RSCFD0CFDRMPTR41 | 00000000 _H | 32 | FFD22524 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 41 | RSCFD0CFDRMFST41 | 00000000 _H | 32 | FFD22528 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 41 | RSCFD0CFDRMDF0_41 | 00000000 _H | 32 | FFD2252C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 41 | RSCFD0CFDRMDF1_41 | 00000000 _H | 32 | FFD22530 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 41 | RSCFD0CFDRMDF2_41 | 00000000 _H | 32 | FFD22534 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 41 | RSCFD0CFDRMDF3_41 | 00000000 _H | 32 | FFD22538 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 41 | RSCFD0CFDRMDF4_41 | 00000000 _H | 32 | FFD2253C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 42 | RSCFD0CFDRMID42 | 00000000 _H | 32 | FFD22540 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 42 | RSCFD0CFDRMPTR42 | 00000000 _H | 32 | FFD22544 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 42 | RSCFD0CFDRMFST42 | 00000000 _H | 32 | FFD22548 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 42 | RSCFD0CFDRMDF0_42 | 00000000 _H | 32 | FFD2254C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 42 | RSCFD0CFDRMDF1_42 | 00000000 _H | 32 | FFD22550 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 42 | RSCFD0CFDRMDF2_42 | 00000000 _H | 32 | FFD22554 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 42 | RSCFD0CFDRMDF3_42 | 00000000 _H | 32 | FFD22558 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 42 | RSCFD0CFDRMDF4_42 | 00000000 _H | 32 | FFD2255C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 43 | RSCFD0CFDRMID43 | 00000000 _H | 32 | FFD22560 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 43 | RSCFD0CFDRMPTR43 | 00000000 _H | 32 | FFD22564 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 43 | RSCFD0CFDRMFST43 | 00000000 _H | 32 | FFD22568 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 43 | RSCFD0CFDRMDF0_43 | 00000000 _H | 32 | FFD2256C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 43 | RSCFD0CFDRMDF1_43 | 00000000 _H | 32 | FFD22570 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 43 | RSCFD0CFDRMDF2_43 | 00000000 _H | 32 | FFD22574 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 43 | RSCFD0CFDRMDF3_43 | 00000000 _H | 32 | FFD22578 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 43 | RSCFD0CFDRMDF4_43 | 00000000 _H | 32 | FFD2257C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 44 | RSCFD0CFDRMID44 | 00000000 _H | 32 | FFD22580 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 44 | RSCFD0CFDRMPTR44 | 00000000 _H | 32 | FFD22584 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 44 | RSCFD0CFDRMFST44 | 00000000 _H | 32 | FFD22588 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 44 | RSCFD0CFDRMDF0_44 | 00000000 _H | 32 | FFD2258C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 44 | RSCFD0CFDRMDF1_44 | 00000000 _H | 32 | FFD22590 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 44 | RSCFD0CFDRMDF2_44 | 00000000 _H | 32 | FFD22594 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 44 | RSCFD0CFDRMDF3_44 | 00000000 _H | 32 | FFD22598 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 44 | RSCFD0CFDRMDF4_44 | 00000000 _H | 32 | FFD2259C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 45 | RSCFD0CFDRMID45 | 00000000 _H | 32 | FFD225A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 45 | RSCFD0CFDRMPTR45 | 00000000 _H | 32 | FFD225A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 45 | RSCFD0CFDRMFST45 | 00000000 _H | 32 | FFD225A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (58/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | RX Message Buffer Data Field 0 Register 45 | RSCFD0CFDRMDF0_45 | 00000000 _H | 32 | FFD225AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 45 | RSCFD0CFDRMDF1_45 | 00000000 _H | 32 | FFD225B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 45 | RSCFD0CFDRMDF2_45 | 00000000 _H | 32 | FFD225B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 45 | RSCFD0CFDRMDF3_45 | 00000000 _H | 32 | FFD225B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 45 | RSCFD0CFDRMDF4_45 | 00000000 _H | 32 | FFD225BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 46 | RSCFD0CFDRMID46 | 00000000 _H | 32 | FFD225C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 46 | RSCFD0CFDRMPTR46 | 00000000 _H | 32 | FFD225C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 46 | RSCFD0CFDRMFMDSTS46 | 00000000 _H | 32 | FFD225C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 46 | RSCFD0CFDRMDF0_46 | 00000000 _H | 32 | FFD225CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 46 | RSCFD0CFDRMDF1_46 | 00000000 _H | 32 | FFD225D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 46 | RSCFD0CFDRMDF2_46 | 00000000 _H | 32 | FFD225D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 46 | RSCFD0CFDRMDF3_46 | 00000000 _H | 32 | FFD225D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 46 | RSCFD0CFDRMDF4_46 | 00000000 _H | 32 | FFD225DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer ID register 47 | RSCFD0CFDRMID47 | 00000000 _H | 32 | FFD225E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive buffer pointer register 47 | RSCFD0CFDRMPTR47 | 00000000 _H | 32 | FFD225E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer CAN FD Status Register 47 | RSCFD0CFDRMFMDSTS47 | 00000000 _H | 32 | FFD225E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 0 Register 47 | RSCFD0CFDRMDF0_47 | 00000000 _H | 32 | FFD225EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 1 Register 47 | RSCFD0CFDRMDF1_47 | 00000000 _H | 32 | FFD225F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 2 Register 47 | RSCFD0CFDRMDF2_47 | 00000000 _H | 32 | FFD225F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 3 Register 47 | RSCFD0CFDRMDF3_47 | 00000000 _H | 32 | FFD225F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RX Message Buffer Data Field 4 Register 47 | RSCFD0CFDRMDF4_47 | 00000000 _H | 32 | FFD225FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer access ID register 0 | RSCFD0CFDRFID0 | 00000000 _H | 32 | FFD23000 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer access pointer register 0 | RSCFD0CFDRFPTR0 | 00000000 _H | 32 | FFD23004 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO CAN FD status register 0 | RSCFD0CFDRFFDSTS0 | 00000000 _H | 32 | FFD23008 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 0 | RSCFD0CFDRFDF0_0 | 00000000 _H | 32 | FFD2300C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 0 | RSCFD0CFDRFDF1_0 | 00000000 _H | 32 | FFD23010 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 0 | RSCFD0CFDRFDF2_0 | 00000000 _H | 32 | FFD23014 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 0 | RSCFD0CFDRFDF3_0 | 00000000 _H | 32 | FFD23018 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 0 | RSCFD0CFDRFDF4_0 | 00000000 _H | 32 | FFD2301C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 0 | RSCFD0CFDRFDF5_0 | 00000000 _H | 32 | FFD23020 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 0 | RSCFD0CFDRFDF6_0 | 00000000 _H | 32 | FFD23024 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 0 | RSCFD0CFDRFDF7_0 | 00000000 _H | 32 | FFD23028 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 0 | RSCFD0CFDRFDF8_0 | 00000000 _H | 32 | FFD2302C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 0 | RSCFD0CFDRFDF9_0 | 00000000 _H | 32 | FFD23030 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 0 | RSCFD0CFDRFDF10_0 | 00000000 _H | 32 | FFD23034 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 0 | RSCFD0CFDRFDF11_0 | 00000000 _H | 32 | FFD23038 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 0 | RSCFD0CFDRFDF12_0 | 00000000 _H | 32 | FFD2303C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 0 | RSCFD0CFDRFDF13_0 | 00000000 _H | 32 | FFD23040 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 0 | RSCFD0CFDRFDF14_0 | 00000000 _H | 32 | FFD23044 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 0 | RSCFD0CFDRFDF15_0 | 00000000 _H | 32 | FFD23048 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer access ID register 1 | RSCFD0CFDRFID1 | 00000000 _H | 32 | FFD23080 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer access pointer register 1 | RSCFD0CFDRFPTR1 | 00000000 _H | 32 | FFD23084 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (59/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Receive FIFO CAN FD status register 1 | RSCFD0CFDRFFDSTS1 | 00000000 _H | 32 | FFD23088 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 1 | RSCFD0CFDRFDF0_1 | 00000000 _H | 32 | FFD2308C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 1 | RSCFD0CFDRFDF1_1 | 00000000 _H | 32 | FFD23090 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 1 | RSCFD0CFDRFDF2_1 | 00000000 _H | 32 | FFD23094 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 1 | RSCFD0CFDRFDF3_1 | 00000000 _H | 32 | FFD23098 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 1 | RSCFD0CFDRFDF4_1 | 00000000 _H | 32 | FFD2309C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 1 | RSCFD0CFDRFDF5_1 | 00000000 _H | 32 | FFD230A0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 1 | RSCFD0CFDRFDF6_1 | 00000000 _H | 32 | FFD230A4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 1 | RSCFD0CFDRFDF7_1 | 00000000 _H | 32 | FFD230A8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 1 | RSCFD0CFDRFDF8_1 | 00000000 _H | 32 | FFD230AC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 1 | RSCFD0CFDRFDF9_1 | 00000000 _H | 32 | FFD230B0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 1 | RSCFD0CFDRFDF10_1 | 00000000 _H | 32 | FFD230B4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 1 | RSCFD0CFDRFDF11_1 | 00000000 _H | 32 | FFD230B8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 1 | RSCFD0CFDRFDF12_1 | 00000000 _H | 32 | FFD230BC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 1 | RSCFD0CFDRFDF13_1 | 00000000 _H | 32 | FFD230C0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 1 | RSCFD0CFDRFDF14_1 | 00000000 _H | 32 | FFD230C4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 1 | RSCFD0CFDRFDF15_1 | 00000000 _H | 32 | FFD230C8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer access ID register 2 | RSCFD0CFDRFID2 | 00000000 _H | 32 | FFD23100 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer access pointer register 2 | RSCFD0CFDRFPTR2 | 00000000 _H | 32 | FFD23104 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO CAN FD status register 2 | RSCFD0CFDRFFDSTS2 | 00000000 _H | 32 | FFD23108 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 2 | RSCFD0CFDRFDF0_2 | 00000000 _H | 32 | FFD2310C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 2 | RSCFD0CFDRFDF1_2 | 00000000 _H | 32 | FFD23110 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 2 | RSCFD0CFDRFDF2_2 | 00000000 _H | 32 | FFD23114 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 2 | RSCFD0CFDRFDF3_2 | 00000000 _H | 32 | FFD23118 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 2 | RSCFD0CFDRFDF4_2 | 00000000 _H | 32 | FFD2311C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 2 | RSCFD0CFDRFDF5_2 | 00000000 _H | 32 | FFD23120 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 2 | RSCFD0CFDRFDF6_2 | 00000000 _H | 32 | FFD23124 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 2 | RSCFD0CFDRFDF7_2 | 00000000 _H | 32 | FFD23128 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 2 | RSCFD0CFDRFDF8_2 | 00000000 _H | 32 | FFD2312C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 2 | RSCFD0CFDRFDF9_2 | 00000000 _H | 32 | FFD23130 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 2 | RSCFD0CFDRFDF10_2 | 00000000 _H | 32 | FFD23134 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 2 | RSCFD0CFDRFDF11_2 | 00000000 _H | 32 | FFD23138 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 2 | RSCFD0CFDRFDF12_2 | 00000000 _H | 32 | FFD2313C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 2 | RSCFD0CFDRFDF13_2 | 00000000 _H | 32 | FFD23140 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 2 | RSCFD0CFDRFDF14_2 | 00000000 _H | 32 | FFD23144 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 2 | RSCFD0CFDRFDF15_2 | 00000000 _H | 32 | FFD23148 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer access ID register 3 | RSCFD0CFDRFID3 | 00000000 _H | 32 | FFD23180 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (60/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Receive FIFO buffer access pointer register 3 | RSCFD0CFDRFPTR3 | 00000000 _H | 32 | FFD23184 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO CAN FD status register 3 | RSCFD0CFDRFFDSTS3 | 00000000 _H | 32 | FFD23188 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 3 | RSCFD0CFDRFDF0_3 | 00000000 _H | 32 | FFD2318C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 3 | RSCFD0CFDRFDF1_3 | 00000000 _H | 32 | FFD23190 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 3 | RSCFD0CFDRFDF2_3 | 00000000 _H | 32 | FFD23194 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 3 | RSCFD0CFDRFDF3_3 | 00000000 _H | 32 | FFD23198 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 3 | RSCFD0CFDRFDF4_3 | 00000000 _H | 32 | FFD2319C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 3 | RSCFD0CFDRFDF5_3 | 00000000 _H | 32 | FFD231A0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 3 | RSCFD0CFDRFDF6_3 | 00000000 _H | 32 | FFD231A4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 3 | RSCFD0CFDRFDF7_3 | 00000000 _H | 32 | FFD231A8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 3 | RSCFD0CFDRFDF8_3 | 00000000 _H | 32 | FFD231AC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 3 | RSCFD0CFDRFDF9_3 | 00000000 _H | 32 | FFD231B0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 3 | RSCFD0CFDRFDF10_3 | 00000000 _H | 32 | FFD231B4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 3 | RSCFD0CFDRFDF11_3 | 00000000 _H | 32 | FFD231B8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 3 | RSCFD0CFDRFDF12_3 | 00000000 _H | 32 | FFD231BC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 3 | RSCFD0CFDRFDF13_3 | 00000000 _H | 32 | FFD231C0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 3 | RSCFD0CFDRFDF14_3 | 00000000 _H | 32 | FFD231C4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 3 | RSCFD0CFDRFDF15_3 | 00000000 _H | 32 | FFD231C8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer access ID register 4 | RSCFD0CFDRFID4 | 00000000 _H | 32 | FFD23200 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer access pointer register 4 | RSCFD0CFDRFPTR4 | 00000000 _H | 32 | FFD23204 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO CAN FD status register 4 | RSCFD0CFDRFFDSTS4 | 00000000 _H | 32 | FFD23208 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 4 | RSCFD0CFDRFDF0_4 | 00000000 _H | 32 | FFD2320C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 4 | RSCFD0CFDRFDF1_4 | 00000000 _H | 32 | FFD23210 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 4 | RSCFD0CFDRFDF2_4 | 00000000 _H | 32 | FFD23214 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 4 | RSCFD0CFDRFDF3_4 | 00000000 _H | 32 | FFD23218 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 4 | RSCFD0CFDRFDF4_4 | 00000000 _H | 32 | FFD2321C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 4 | RSCFD0CFDRFDF5_4 | 00000000 _H | 32 | FFD23220 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 4 | RSCFD0CFDRFDF6_4 | 00000000 _H | 32 | FFD23224 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 4 | RSCFD0CFDRFDF7_4 | 00000000 _H | 32 | FFD23228 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 4 | RSCFD0CFDRFDF8_4 | 00000000 _H | 32 | FFD2322C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 4 | RSCFD0CFDRFDF9_4 | 00000000 _H | 32 | FFD23230 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 4 | RSCFD0CFDRFDF10_4 | 00000000 _H | 32 | FFD23234 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 4 | RSCFD0CFDRFDF11_4 | 00000000 _H | 32 | FFD23238 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 4 | RSCFD0CFDRFDF12_4 | 00000000 _H | 32 | FFD2323C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 4 | RSCFD0CFDRFDF13_4 | 00000000 _H | 32 | FFD23240 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 4 | RSCFD0CFDRFDF14_4 | 00000000 _H | 32 | FFD23244 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 4 | RSCFD0CFDRFDF15_4 | 00000000 _H | 32 | FFD23248 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (61/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Receive FIFO buffer access ID register 5 | RSCFD0CFDRFID5 | 00000000 _H | 32 | FFD23280 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer access pointer register 5 | RSCFD0CFDRFPTR5 | 00000000 _H | 32 | FFD23284 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO CAN FD status register 5 | RSCFD0CFDRFFDSTS5 | 00000000 _H | 32 | FFD23288 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 5 | RSCFD0CFDRFDF0_5 | 00000000 _H | 32 | FFD2328C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 5 | RSCFD0CFDRFDF1_5 | 00000000 _H | 32 | FFD23290 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 5 | RSCFD0CFDRFDF2_5 | 00000000 _H | 32 | FFD23294 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 5 | RSCFD0CFDRFDF3_5 | 00000000 _H | 32 | FFD23298 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 5 | RSCFD0CFDRFDF4_5 | 00000000 _H | 32 | FFD2329C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 5 | RSCFD0CFDRFDF5_5 | 00000000 _H | 32 | FFD232A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 5 | RSCFD0CFDRFDF6_5 | 00000000 _H | 32 | FFD232A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 5 | RSCFD0CFDRFDF7_5 | 00000000 _H | 32 | FFD232A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 5 | RSCFD0CFDRFDF8_5 | 00000000 _H | 32 | FFD232AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 5 | RSCFD0CFDRFDF9_5 | 00000000 _H | 32 | FFD232B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 5 | RSCFD0CFDRFDF10_5 | 00000000 _H | 32 | FFD232B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 5 | RSCFD0CFDRFDF11_5 | 00000000 _H | 32 | FFD232B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 5 | RSCFD0CFDRFDF12_5 | 00000000 _H | 32 | FFD232BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 5 | RSCFD0CFDRFDF13_5 | 00000000 _H | 32 | FFD232C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 5 | RSCFD0CFDRFDF14_5 | 00000000 _H | 32 | FFD232C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 5 | RSCFD0CFDRFDF15_5 | 00000000 _H | 32 | FFD232C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer access ID register 6 | RSCFD0CFDRFID6 | 00000000 _H | 32 | FFD23300 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO buffer access pointer register 6 | RSCFD0CFDRFPTR6 | 00000000 _H | 32 | FFD23304 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO CAN FD status register 6 | RSCFD0CFDRFFDSTS6 | 00000000 _H | 32 | FFD23308 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 6 | RSCFD0CFDRFDF0_6 | 00000000 _H | 32 | FFD2330C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 6 | RSCFD0CFDRFDF1_6 | 00000000 _H | 32 | FFD23310 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 6 | RSCFD0CFDRFDF2_6 | 00000000 _H | 32 | FFD23314 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 6 | RSCFD0CFDRFDF3_6 | 00000000 _H | 32 | FFD23318 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 6 | RSCFD0CFDRFDF4_6 | 00000000 _H | 32 | FFD2331C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 6 | RSCFD0CFDRFDF5_6 | 00000000 _H | 32 | FFD23320 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 6 | RSCFD0CFDRFDF6_6 | 00000000 _H | 32 | FFD23324 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 6 | RSCFD0CFDRFDF7_6 | 00000000 _H | 32 | FFD23328 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 6 | RSCFD0CFDRFDF8_6 | 00000000 _H | 32 | FFD2332C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 6 | RSCFD0CFDRFDF9_6 | 00000000 _H | 32 | FFD23330 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 6 | RSCFD0CFDRFDF10_6 | 00000000 _H | 32 | FFD23334 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 6 | RSCFD0CFDRFDF11_6 | 00000000 _H | 32 | FFD23338 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 6 | RSCFD0CFDRFDF12_6 | 00000000 _H | 32 | FFD2333C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 6 | RSCFD0CFDRFDF13_6 | 00000000 _H | 32 | FFD23340 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 6 | RSCFD0CFDRFDF14_6 | 00000000 _H | 32 | FFD23344 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (62/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 6 | RSCFD0CFDRFDF15_6 | 00000000 _H | 32 | FFD23348 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer access ID register 7 | RSCFD0CFDRFID7 | 00000000 _H | 32 | FFD23380 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO buffer access pointer register 7 | RSCFD0CFDRFPTR7 | 00000000 _H | 32 | FFD23384 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO CAN FD status register 7 | RSCFD0CFDRFFDSTS7 | 00000000 _H | 32 | FFD23388 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 0 Register 7 | RSCFD0CFDRFDF0_7 | 00000000 _H | 32 | FFD2338C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 1 Register 7 | RSCFD0CFDRFDF1_7 | 00000000 _H | 32 | FFD23390 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 2 Register 7 | RSCFD0CFDRFDF2_7 | 00000000 _H | 32 | FFD23394 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 3 Register 7 | RSCFD0CFDRFDF3_7 | 00000000 _H | 32 | FFD23398 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 4 Register 7 | RSCFD0CFDRFDF4_7 | 00000000 _H | 32 | FFD2339C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 5 Register 7 | RSCFD0CFDRFDF5_7 | 00000000 _H | 32 | FFD233A0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 6 Register 7 | RSCFD0CFDRFDF6_7 | 00000000 _H | 32 | FFD233A4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 7 Register 7 | RSCFD0CFDRFDF7_7 | 00000000 _H | 32 | FFD233A8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 8 Register 7 | RSCFD0CFDRFDF8_7 | 00000000 _H | 32 | FFD233AC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 9 Register 7 | RSCFD0CFDRFDF9_7 | 00000000 _H | 32 | FFD233B0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 10 Register 7 | RSCFD0CFDRFDF10_7 | 00000000 _H | 32 | FFD233B4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 11 Register 7 | RSCFD0CFDRFDF11_7 | 00000000 _H | 32 | FFD233B8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 12 Register 7 | RSCFD0CFDRFDF12_7 | 00000000 _H | 32 | FFD233BC _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 13 Register 7 | RSCFD0CFDRFDF13_7 | 00000000 _H | 32 | FFD233C0 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 14 Register 7 | RSCFD0CFDRFDF14_7 | 00000000 _H | 32 | FFD233C4 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Receive FIFO Buffer Access Data Field 15 Register 7 | RSCFD0CFDRFDF15_7 | 00000000 _H | 32 | FFD233C8 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 0 | RSCFD0CFDCFID0 | 00000000 _H | 32 | FFD23400 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 0 | RSCFD0CFDCFPTR0 | 00000000 _H | 32 | FFD23404 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 0 | RSCFD0CFDCFFDSTS0 | 00000000 _H | 32 | FFD23408 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 0 | RSCFD0CFDCFDF0_0 | 00000000 _H | 32 | FFD2340C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 0 | RSCFD0CFDCFDF1_0 | 00000000 _H | 32 | FFD23410 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 0 | RSCFD0CFDCFDF2_0 | 00000000 _H | 32 | FFD23414 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 0 | RSCFD0CFDCFDF3_0 | 00000000 _H | 32 | FFD23418 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 0 | RSCFD0CFDCFDF4_0 | 00000000 _H | 32 | FFD2341C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 0 | RSCFD0CFDCFDF5_0 | 00000000 _H | 32 | FFD23420 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 0 | RSCFD0CFDCFDF6_0 | 00000000 _H | 32 | FFD23424 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 0 | RSCFD0CFDCFDF7_0 | 00000000 _H | 32 | FFD23428 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 0 | RSCFD0CFDCFDF8_0 | 00000000 _H | 32 | FFD2342C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 0 | RSCFD0CFDCFDF9_0 | 00000000 _H | 32 | FFD23430 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 0 | RSCFD0CFDCFDF10_0 | 00000000 _H | 32 | FFD23434 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 0 | RSCFD0CFDCFDF11_0 | 00000000 _H | 32 | FFD23438 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 0 | RSCFD0CFDCFDF12_0 | 00000000 _H | 32 | FFD2343C _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 0 | RSCFD0CFDCFDF13_0 | 00000000 _H | 32 | FFD23440 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (63/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 0 | RSCFD0CFDCDFD14_0 | 00000000 _H | 32 | FFD23444 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 0 | RSCFD0CFDCDFD15_0 | 00000000 _H | 32 | FFD23448 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 1 | RSCFD0CFDCFD1 | 00000000 _H | 32 | FFD23480 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 1 | RSCFD0CFDCFPTR1 | 00000000 _H | 32 | FFD23484 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 1 | RSCFD0CFDCFFDCSTS1 | 00000000 _H | 32 | FFD23488 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 1 | RSCFD0CFDCDFD0_1 | 00000000 _H | 32 | FFD2348C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 1 | RSCFD0CFDCDFD1_1 | 00000000 _H | 32 | FFD23490 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 1 | RSCFD0CFDCDFD2_1 | 00000000 _H | 32 | FFD23494 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 1 | RSCFD0CFDCDFD3_1 | 00000000 _H | 32 | FFD23498 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 1 | RSCFD0CFDCDFD4_1 | 00000000 _H | 32 | FFD2349C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 1 | RSCFD0CFDCDFD5_1 | 00000000 _H | 32 | FFD234A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 1 | RSCFD0CFDCDFD6_1 | 00000000 _H | 32 | FFD234A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 1 | RSCFD0CFDCDFD7_1 | 00000000 _H | 32 | FFD234A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 1 | RSCFD0CFDCDFD8_1 | 00000000 _H | 32 | FFD234AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 1 | RSCFD0CFDCDFD9_1 | 00000000 _H | 32 | FFD234B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 1 | RSCFD0CFDCDFD10_1 | 00000000 _H | 32 | FFD234B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 1 | RSCFD0CFDCDFD11_1 | 00000000 _H | 32 | FFD234B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 1 | RSCFD0CFDCDFD12_1 | 00000000 _H | 32 | FFD234BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 1 | RSCFD0CFDCDFD13_1 | 00000000 _H | 32 | FFD234C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 1 | RSCFD0CFDCDFD14_1 | 00000000 _H | 32 | FFD234C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 1 | RSCFD0CFDCDFD15_1 | 00000000 _H | 32 | FFD234C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 2 | RSCFD0CFDCFD2 | 00000000 _H | 32 | FFD23500 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 2 | RSCFD0CFDCFPTR2 | 00000000 _H | 32 | FFD23504 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 2 | RSCFD0CFDCFFDCSTS2 | 00000000 _H | 32 | FFD23508 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 2 | RSCFD0CFDCDFD0_2 | 00000000 _H | 32 | FFD2350C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 2 | RSCFD0CFDCDFD1_2 | 00000000 _H | 32 | FFD23510 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 2 | RSCFD0CFDCDFD2_2 | 00000000 _H | 32 | FFD23514 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 2 | RSCFD0CFDCDFD3_2 | 00000000 _H | 32 | FFD23518 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 2 | RSCFD0CFDCDFD4_2 | 00000000 _H | 32 | FFD2351C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 2 | RSCFD0CFDCDFD5_2 | 00000000 _H | 32 | FFD23520 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 2 | RSCFD0CFDCDFD6_2 | 00000000 _H | 32 | FFD23524 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 2 | RSCFD0CFDCDFD7_2 | 00000000 _H | 32 | FFD23528 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 2 | RSCFD0CFDCDFD8_2 | 00000000 _H | 32 | FFD2352C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 2 | RSCFD0CFDCDFD9_2 | 00000000 _H | 32 | FFD23530 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 2 | RSCFD0CFDCDFD10_2 | 00000000 _H | 32 | FFD23534 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 2 | RSCFD0CFDCDFD11_2 | 00000000 _H | 32 | FFD23538 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (64/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 2 | RSCFD0CFDCFD12_2 | 00000000 _H | 32 | FFD2353C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 2 | RSCFD0CFDCFD13_2 | 00000000 _H | 32 | FFD23540 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 2 | RSCFD0CFDCFD14_2 | 00000000 _H | 32 | FFD23544 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 2 | RSCFD0CFDCFD15_2 | 00000000 _H | 32 | FFD23548 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 3 | RSCFD0CFDCFD3 | 00000000 _H | 32 | FFD23580 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 3 | RSCFD0CFDCFPTR3 | 00000000 _H | 32 | FFD23584 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 3 | RSCFD0CFDCFFDCSTS3 | 00000000 _H | 32 | FFD23588 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 3 | RSCFD0CFDCFD0_3 | 00000000 _H | 32 | FFD2358C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 3 | RSCFD0CFDCFD1_3 | 00000000 _H | 32 | FFD23590 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 3 | RSCFD0CFDCFD2_3 | 00000000 _H | 32 | FFD23594 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 3 | RSCFD0CFDCFD3_3 | 00000000 _H | 32 | FFD23598 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 3 | RSCFD0CFDCFD4_3 | 00000000 _H | 32 | FFD2359C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 3 | RSCFD0CFDCFD5_3 | 00000000 _H | 32 | FFD235A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 3 | RSCFD0CFDCFD6_3 | 00000000 _H | 32 | FFD235A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 3 | RSCFD0CFDCFD7_3 | 00000000 _H | 32 | FFD235A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 3 | RSCFD0CFDCFD8_3 | 00000000 _H | 32 | FFD235AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 3 | RSCFD0CFDCFD9_3 | 00000000 _H | 32 | FFD235B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 3 | RSCFD0CFDCFD10_3 | 00000000 _H | 32 | FFD235B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 3 | RSCFD0CFDCFD11_3 | 00000000 _H | 32 | FFD235B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 3 | RSCFD0CFDCFD12_3 | 00000000 _H | 32 | FFD235BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 3 | RSCFD0CFDCFD13_3 | 00000000 _H | 32 | FFD235C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 3 | RSCFD0CFDCFD14_3 | 00000000 _H | 32 | FFD235C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 3 | RSCFD0CFDCFD15_3 | 00000000 _H | 32 | FFD235C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 4 | RSCFD0CFDCFD4 | 00000000 _H | 32 | FFD23600 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 4 | RSCFD0CFDCFPTR4 | 00000000 _H | 32 | FFD23604 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 4 | RSCFD0CFDCFFDCSTS4 | 00000000 _H | 32 | FFD23608 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 4 | RSCFD0CFDCFD0_4 | 00000000 _H | 32 | FFD2360C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 4 | RSCFD0CFDCFD1_4 | 00000000 _H | 32 | FFD23610 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 4 | RSCFD0CFDCFD2_4 | 00000000 _H | 32 | FFD23614 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 4 | RSCFD0CFDCFD3_4 | 00000000 _H | 32 | FFD23618 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 4 | RSCFD0CFDCFD4_4 | 00000000 _H | 32 | FFD2361C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 4 | RSCFD0CFDCFD5_4 | 00000000 _H | 32 | FFD23620 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 4 | RSCFD0CFDCFD6_4 | 00000000 _H | 32 | FFD23624 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 4 | RSCFD0CFDCFD7_4 | 00000000 _H | 32 | FFD23628 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 4 | RSCFD0CFDCFD8_4 | 00000000 _H | 32 | FFD2362C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 4 | RSCFD0CFDCFD9_4 | 00000000 _H | 32 | FFD23630 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (65/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 4 | RSCFD0CFDCDFD10_4 | 00000000 _H | 32 | FFD23634 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 4 | RSCFD0CFDCDFD11_4 | 00000000 _H | 32 | FFD23638 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 4 | RSCFD0CFDCDFD12_4 | 00000000 _H | 32 | FFD2363C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 4 | RSCFD0CFDCDFD13_4 | 00000000 _H | 32 | FFD23640 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 4 | RSCFD0CFDCDFD14_4 | 00000000 _H | 32 | FFD23644 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 4 | RSCFD0CFDCDFD15_4 | 00000000 _H | 32 | FFD23648 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 5 | RSCFD0CFDCFD5 | 00000000 _H | 32 | FFD23680 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 5 | RSCFD0CFDCFPTR5 | 00000000 _H | 32 | FFD23684 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 5 | RSCFD0CFDCFFDCSTS5 | 00000000 _H | 32 | FFD23688 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 5 | RSCFD0CFDCDFD0_5 | 00000000 _H | 32 | FFD2368C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 5 | RSCFD0CFDCDFD1_5 | 00000000 _H | 32 | FFD23690 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 5 | RSCFD0CFDCDFD2_5 | 00000000 _H | 32 | FFD23694 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 5 | RSCFD0CFDCDFD3_5 | 00000000 _H | 32 | FFD23698 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 5 | RSCFD0CFDCDFD4_5 | 00000000 _H | 32 | FFD2369C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 5 | RSCFD0CFDCDFD5_5 | 00000000 _H | 32 | FFD236A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 5 | RSCFD0CFDCDFD6_5 | 00000000 _H | 32 | FFD236A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 5 | RSCFD0CFDCDFD7_5 | 00000000 _H | 32 | FFD236A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 5 | RSCFD0CFDCDFD8_5 | 00000000 _H | 32 | FFD236AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 5 | RSCFD0CFDCDFD9_5 | 00000000 _H | 32 | FFD236B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 5 | RSCFD0CFDCDFD10_5 | 00000000 _H | 32 | FFD236B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 5 | RSCFD0CFDCDFD11_5 | 00000000 _H | 32 | FFD236B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 5 | RSCFD0CFDCDFD12_5 | 00000000 _H | 32 | FFD236BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 5 | RSCFD0CFDCDFD13_5 | 00000000 _H | 32 | FFD236C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 5 | RSCFD0CFDCDFD14_5 | 00000000 _H | 32 | FFD236C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 5 | RSCFD0CFDCDFD15_5 | 00000000 _H | 32 | FFD236C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 6 | RSCFD0CFDCFD6 | 00000000 _H | 32 | FFD23700 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 6 | RSCFD0CFDCFPTR6 | 00000000 _H | 32 | FFD23704 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 6 | RSCFD0CFDCFFDCSTS6 | 00000000 _H | 32 | FFD23708 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 6 | RSCFD0CFDCDFD0_6 | 00000000 _H | 32 | FFD2370C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 6 | RSCFD0CFDCDFD1_6 | 00000000 _H | 32 | FFD23710 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 6 | RSCFD0CFDCDFD2_6 | 00000000 _H | 32 | FFD23714 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 6 | RSCFD0CFDCDFD3_6 | 00000000 _H | 32 | FFD23718 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 6 | RSCFD0CFDCDFD4_6 | 00000000 _H | 32 | FFD2371C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 6 | RSCFD0CFDCDFD5_6 | 00000000 _H | 32 | FFD23720 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 6 | RSCFD0CFDCDFD6_6 | 00000000 _H | 32 | FFD23724 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 6 | RSCFD0CFDCDFD7_6 | 00000000 _H | 32 | FFD23728 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (66/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 6 | RSCFD0CFDCDFD8_6 | 00000000 _H | 32 | FFD2372C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 6 | RSCFD0CFDCDFD9_6 | 00000000 _H | 32 | FFD23730 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 6 | RSCFD0CFDCDFD10_6 | 00000000 _H | 32 | FFD23734 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 6 | RSCFD0CFDCDFD11_6 | 00000000 _H | 32 | FFD23738 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 6 | RSCFD0CFDCDFD12_6 | 00000000 _H | 32 | FFD2373C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 6 | RSCFD0CFDCDFD13_6 | 00000000 _H | 32 | FFD23740 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 6 | RSCFD0CFDCDFD14_6 | 00000000 _H | 32 | FFD23744 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 6 | RSCFD0CFDCDFD15_6 | 00000000 _H | 32 | FFD23748 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 7 | RSCFD0CFDCFD7 | 00000000 _H | 32 | FFD23780 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 7 | RSCFD0CFDCFPTR7 | 00000000 _H | 32 | FFD23784 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 7 | RSCFD0CFDCFFDCSTS7 | 00000000 _H | 32 | FFD23788 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 7 | RSCFD0CFDCDFD0_7 | 00000000 _H | 32 | FFD2378C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 7 | RSCFD0CFDCDFD1_7 | 00000000 _H | 32 | FFD23790 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 7 | RSCFD0CFDCDFD2_7 | 00000000 _H | 32 | FFD23794 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 7 | RSCFD0CFDCDFD3_7 | 00000000 _H | 32 | FFD23798 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 7 | RSCFD0CFDCDFD4_7 | 00000000 _H | 32 | FFD2379C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 7 | RSCFD0CFDCDFD5_7 | 00000000 _H | 32 | FFD237A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 7 | RSCFD0CFDCDFD6_7 | 00000000 _H | 32 | FFD237A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 7 | RSCFD0CFDCDFD7_7 | 00000000 _H | 32 | FFD237A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 7 | RSCFD0CFDCDFD8_7 | 00000000 _H | 32 | FFD237AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 7 | RSCFD0CFDCDFD9_7 | 00000000 _H | 32 | FFD237B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 7 | RSCFD0CFDCDFD10_7 | 00000000 _H | 32 | FFD237B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 7 | RSCFD0CFDCDFD11_7 | 00000000 _H | 32 | FFD237B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 7 | RSCFD0CFDCDFD12_7 | 00000000 _H | 32 | FFD237BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 7 | RSCFD0CFDCDFD13_7 | 00000000 _H | 32 | FFD237C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 7 | RSCFD0CFDCDFD14_7 | 00000000 _H | 32 | FFD237C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 7 | RSCFD0CFDCDFD15_7 | 00000000 _H | 32 | FFD237C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access ID register 8 | RSCFD0CFDCFD8 | 00000000 _H | 32 | FFD23800 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO buffer access pointer register 8 | RSCFD0CFDCFPTR8 | 00000000 _H | 32 | FFD23804 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO CAN FD configuration/status register 8 | RSCFD0CFDCFFDCSTS8 | 00000000 _H | 32 | FFD23808 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 0 Register 8 | RSCFD0CFDCDFD0_8 | 00000000 _H | 32 | FFD2380C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 1 Register 8 | RSCFD0CFDCDFD1_8 | 00000000 _H | 32 | FFD23810 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 2 Register 8 | RSCFD0CFDCDFD2_8 | 00000000 _H | 32 | FFD23814 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 3 Register 8 | RSCFD0CFDCDFD3_8 | 00000000 _H | 32 | FFD23818 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 4 Register 8 | RSCFD0CFDCDFD4_8 | 00000000 _H | 32 | FFD2381C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 5 Register 8 | RSCFD0CFDCDFD5_8 | 00000000 _H | 32 | FFD23820 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (67/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 6 Register 8 | RSCFD0CFDCDF6_8 | 00000000 _H | 32 | FFD23824 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 7 Register 8 | RSCFD0CFDCDF7_8 | 00000000 _H | 32 | FFD23828 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 8 Register 8 | RSCFD0CFDCDF8_8 | 00000000 _H | 32 | FFD2382C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 9 Register 8 | RSCFD0CFDCDF9_8 | 00000000 _H | 32 | FFD23830 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 10 Register 8 | RSCFD0CFDCDF10_8 | 00000000 _H | 32 | FFD23834 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 11 Register 8 | RSCFD0CFDCDF11_8 | 00000000 _H | 32 | FFD23838 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 12 Register 8 | RSCFD0CFDCDF12_8 | 00000000 _H | 32 | FFD2383C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 13 Register 8 | RSCFD0CFDCDF13_8 | 00000000 _H | 32 | FFD23840 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 14 Register 8 | RSCFD0CFDCDF14_8 | 00000000 _H | 32 | FFD23844 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit/receive FIFO Buffer Access Data Field 15 Register 8 | RSCFD0CFDCDF15_8 | 00000000 _H | 32 | FFD23848 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD24000 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD24004 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD24008 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD2400C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD24010 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD24014 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD24018 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 0 | RSCFD0CFDTPTR0 | 00000000 _H | 32 | FFD2401C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD24020 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD24024 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD24028 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD2402C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD24030 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD24034 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD24038 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 1 | RSCFD0CFDTPTR1 | 00000000 _H | 32 | FFD2403C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD24040 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD24044 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD24048 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD2404C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD24050 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD24054 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD24058 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 2 | RSCFD0CFDTPTR2 | 00000000 _H | 32 | FFD2405C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD24060 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD24064 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD24068 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD2406C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD24070 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD24074 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD24078 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 3 | RSCFD0CFDTPTR3 | 00000000 _H | 32 | FFD2407C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 4 | RSCFD0CFDTPTR4 | 00000000 _H | 32 | FFD24080 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 4 | RSCFD0CFDTPTR4 | 00000000 _H | 32 | FFD24084 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (68/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 4 | RSCFD0CFDTMFDCTR4 | 00000000 _H | 32 | FFD24088 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 4 | RSCFD0CFDTMDF0_4 | 00000000 _H | 32 | FFD2408C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 4 | RSCFD0CFDTMDF1_4 | 00000000 _H | 32 | FFD24090 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 4 | RSCFD0CFDTMDF2_4 | 00000000 _H | 32 | FFD24094 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 4 | RSCFD0CFDTMDF3_4 | 00000000 _H | 32 | FFD24098 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 4 | RSCFD0CFDTMDF4_4 | 00000000 _H | 32 | FFD2409C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 5 | RSCFD0CFDTMID5 | 00000000 _H | 32 | FFD240A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 5 | RSCFD0CFDTMPTR5 | 00000000 _H | 32 | FFD240A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 5 | RSCFD0CFDTMFDCTR5 | 00000000 _H | 32 | FFD240A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 5 | RSCFD0CFDTMDF0_5 | 00000000 _H | 32 | FFD240AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 5 | RSCFD0CFDTMDF1_5 | 00000000 _H | 32 | FFD240B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 5 | RSCFD0CFDTMDF2_5 | 00000000 _H | 32 | FFD240B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 5 | RSCFD0CFDTMDF3_5 | 00000000 _H | 32 | FFD240B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 5 | RSCFD0CFDTMDF4_5 | 00000000 _H | 32 | FFD240BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 6 | RSCFD0CFDTMID6 | 00000000 _H | 32 | FFD240C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 6 | RSCFD0CFDTMPTR6 | 00000000 _H | 32 | FFD240C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 6 | RSCFD0CFDTMFDCTR6 | 00000000 _H | 32 | FFD240C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 6 | RSCFD0CFDTMDF0_6 | 00000000 _H | 32 | FFD240CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 6 | RSCFD0CFDTMDF1_6 | 00000000 _H | 32 | FFD240D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 6 | RSCFD0CFDTMDF2_6 | 00000000 _H | 32 | FFD240D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 6 | RSCFD0CFDTMDF3_6 | 00000000 _H | 32 | FFD240D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 6 | RSCFD0CFDTMDF4_6 | 00000000 _H | 32 | FFD240DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 7 | RSCFD0CFDTMID7 | 00000000 _H | 32 | FFD240E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 7 | RSCFD0CFDTMPTR7 | 00000000 _H | 32 | FFD240E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 7 | RSCFD0CFDTMFDCTR7 | 00000000 _H | 32 | FFD240E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 7 | RSCFD0CFDTMDF0_7 | 00000000 _H | 32 | FFD240EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 7 | RSCFD0CFDTMDF1_7 | 00000000 _H | 32 | FFD240F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 7 | RSCFD0CFDTMDF2_7 | 00000000 _H | 32 | FFD240F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 7 | RSCFD0CFDTMDF3_7 | 00000000 _H | 32 | FFD240F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 7 | RSCFD0CFDTMDF4_7 | 00000000 _H | 32 | FFD240FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 8 | RSCFD0CFDTMID8 | 00000000 _H | 32 | FFD24100 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 8 | RSCFD0CFDTMPTR8 | 00000000 _H | 32 | FFD24104 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 8 | RSCFD0CFDTMFDCTR8 | 00000000 _H | 32 | FFD24108 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 8 | RSCFD0CFDTMDF0_8 | 00000000 _H | 32 | FFD2410C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 8 | RSCFD0CFDTMDF1_8 | 00000000 _H | 32 | FFD24110 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 8 | RSCFD0CFDTMDF2_8 | 00000000 _H | 32 | FFD24114 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 8 | RSCFD0CFDTMDF3_8 | 00000000 _H | 32 | FFD24118 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 8 | RSCFD0CFDTMDF4_8 | 00000000 _H | 32 | FFD2411C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 9 | RSCFD0CFDTMID9 | 00000000 _H | 32 | FFD24120 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 9 | RSCFD0CFDTMPTR9 | 00000000 _H | 32 | FFD24124 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 9 | RSCFD0CFDTMFDCTR9 | 00000000 _H | 32 | FFD24128 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 9 | RSCFD0CFDTMDF0_9 | 00000000 _H | 32 | FFD2412C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 9 | RSCFD0CFDTMDF1_9 | 00000000 _H | 32 | FFD24130 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 9 | RSCFD0CFDTMDF2_9 | 00000000 _H | 32 | FFD24134 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 9 | RSCFD0CFDTMDF3_9 | 00000000 _H | 32 | FFD24138 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 9 | RSCFD0CFDTMDF4_9 | 00000000 _H | 32 | FFD2413C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 10 | RSCFD0CFDTMID10 | 00000000 _H | 32 | FFD24140 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 10 | RSCFD0CFDTMPTR10 | 00000000 _H | 32 | FFD24144 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (69/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 10 | RSCFD0CFDTMFDCTR10 | 00000000 _H | 32 | FFD24148 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 10 | RSCFD0CFDTMDF0_10 | 00000000 _H | 32 | FFD2414C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 10 | RSCFD0CFDTMDF1_10 | 00000000 _H | 32 | FFD24150 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 10 | RSCFD0CFDTMDF2_10 | 00000000 _H | 32 | FFD24154 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 10 | RSCFD0CFDTMDF3_10 | 00000000 _H | 32 | FFD24158 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 10 | RSCFD0CFDTMDF4_10 | 00000000 _H | 32 | FFD2415C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 11 | RSCFD0CFDTMID11 | 00000000 _H | 32 | FFD24160 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 11 | RSCFD0CFDTMPTR11 | 00000000 _H | 32 | FFD24164 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 11 | RSCFD0CFDTMFDCTR11 | 00000000 _H | 32 | FFD24168 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 11 | RSCFD0CFDTMDF0_11 | 00000000 _H | 32 | FFD2416C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 11 | RSCFD0CFDTMDF1_11 | 00000000 _H | 32 | FFD24170 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 11 | RSCFD0CFDTMDF2_11 | 00000000 _H | 32 | FFD24174 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 11 | RSCFD0CFDTMDF3_11 | 00000000 _H | 32 | FFD24178 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 11 | RSCFD0CFDTMDF4_11 | 00000000 _H | 32 | FFD2417C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 12 | RSCFD0CFDTMID12 | 00000000 _H | 32 | FFD24180 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 12 | RSCFD0CFDTMPTR12 | 00000000 _H | 32 | FFD24184 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 12 | RSCFD0CFDTMFDCTR12 | 00000000 _H | 32 | FFD24188 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 12 | RSCFD0CFDTMDF0_12 | 00000000 _H | 32 | FFD2418C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 12 | RSCFD0CFDTMDF1_12 | 00000000 _H | 32 | FFD24190 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 12 | RSCFD0CFDTMDF2_12 | 00000000 _H | 32 | FFD24194 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 12 | RSCFD0CFDTMDF3_12 | 00000000 _H | 32 | FFD24198 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 12 | RSCFD0CFDTMDF4_12 | 00000000 _H | 32 | FFD2419C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 13 | RSCFD0CFDTMID13 | 00000000 _H | 32 | FFD241A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 13 | RSCFD0CFDTMPTR13 | 00000000 _H | 32 | FFD241A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 13 | RSCFD0CFDTMFDCTR13 | 00000000 _H | 32 | FFD241A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 13 | RSCFD0CFDTMDF0_13 | 00000000 _H | 32 | FFD241AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 13 | RSCFD0CFDTMDF1_13 | 00000000 _H | 32 | FFD241B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 13 | RSCFD0CFDTMDF2_13 | 00000000 _H | 32 | FFD241B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 13 | RSCFD0CFDTMDF3_13 | 00000000 _H | 32 | FFD241B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 13 | RSCFD0CFDTMDF4_13 | 00000000 _H | 32 | FFD241BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 14 | RSCFD0CFDTMID14 | 00000000 _H | 32 | FFD241C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 14 | RSCFD0CFDTMPTR14 | 00000000 _H | 32 | FFD241C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 14 | RSCFD0CFDTMFDCTR14 | 00000000 _H | 32 | FFD241C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 14 | RSCFD0CFDTMDF0_14 | 00000000 _H | 32 | FFD241CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 14 | RSCFD0CFDTMDF1_14 | 00000000 _H | 32 | FFD241D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 14 | RSCFD0CFDTMDF2_14 | 00000000 _H | 32 | FFD241D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 14 | RSCFD0CFDTMDF3_14 | 00000000 _H | 32 | FFD241D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 14 | RSCFD0CFDTMDF4_14 | 00000000 _H | 32 | FFD241DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 15 | RSCFD0CFDTMID15 | 00000000 _H | 32 | FFD241E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 15 | RSCFD0CFDTMPTR15 | 00000000 _H | 32 | FFD241E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 15 | RSCFD0CFDTMFDCTR15 | 00000000 _H | 32 | FFD241E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 15 | RSCFD0CFDTMDF0_15 | 00000000 _H | 32 | FFD241EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 15 | RSCFD0CFDTMDF1_15 | 00000000 _H | 32 | FFD241F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 15 | RSCFD0CFDTMDF2_15 | 00000000 _H | 32 | FFD241F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 15 | RSCFD0CFDTMDF3_15 | 00000000 _H | 32 | FFD241F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 15 | RSCFD0CFDTMDF4_15 | 00000000 _H | 32 | FFD241FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 16 | RSCFD0CFDTMID16 | 00000000 _H | 32 | FFD24200 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 16 | RSCFD0CFDTMPTR16 | 00000000 _H | 32 | FFD24204 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (70/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 16 | RSCFD0CFDTMFDCTR16 | 00000000 _H | 32 | FFD24208 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 16 | RSCFD0CFDTMDF0_16 | 00000000 _H | 32 | FFD2420C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 16 | RSCFD0CFDTMDF1_16 | 00000000 _H | 32 | FFD24210 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 16 | RSCFD0CFDTMDF2_16 | 00000000 _H | 32 | FFD24214 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 16 | RSCFD0CFDTMDF3_16 | 00000000 _H | 32 | FFD24218 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 16 | RSCFD0CFDTMDF4_16 | 00000000 _H | 32 | FFD2421C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 17 | RSCFD0CFDTMID17 | 00000000 _H | 32 | FFD24220 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 17 | RSCFD0CFDTMPTR17 | 00000000 _H | 32 | FFD24224 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 17 | RSCFD0CFDTMFDCTR17 | 00000000 _H | 32 | FFD24228 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 17 | RSCFD0CFDTMDF0_17 | 00000000 _H | 32 | FFD2422C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 17 | RSCFD0CFDTMDF1_17 | 00000000 _H | 32 | FFD24230 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 17 | RSCFD0CFDTMDF2_17 | 00000000 _H | 32 | FFD24234 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 17 | RSCFD0CFDTMDF3_17 | 00000000 _H | 32 | FFD24238 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 17 | RSCFD0CFDTMDF4_17 | 00000000 _H | 32 | FFD2423C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 18 | RSCFD0CFDTMID18 | 00000000 _H | 32 | FFD24240 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 18 | RSCFD0CFDTMPTR18 | 00000000 _H | 32 | FFD24244 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 18 | RSCFD0CFDTMFDCTR18 | 00000000 _H | 32 | FFD24248 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 18 | RSCFD0CFDTMDF0_18 | 00000000 _H | 32 | FFD2424C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 18 | RSCFD0CFDTMDF1_18 | 00000000 _H | 32 | FFD24250 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 18 | RSCFD0CFDTMDF2_18 | 00000000 _H | 32 | FFD24254 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 18 | RSCFD0CFDTMDF3_18 | 00000000 _H | 32 | FFD24258 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 18 | RSCFD0CFDTMDF4_18 | 00000000 _H | 32 | FFD2425C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 19 | RSCFD0CFDTMID19 | 00000000 _H | 32 | FFD24260 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 19 | RSCFD0CFDTMPTR19 | 00000000 _H | 32 | FFD24264 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 19 | RSCFD0CFDTMFDCTR19 | 00000000 _H | 32 | FFD24268 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 19 | RSCFD0CFDTMDF0_19 | 00000000 _H | 32 | FFD2426C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 19 | RSCFD0CFDTMDF1_19 | 00000000 _H | 32 | FFD24270 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 19 | RSCFD0CFDTMDF2_19 | 00000000 _H | 32 | FFD24274 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 19 | RSCFD0CFDTMDF3_19 | 00000000 _H | 32 | FFD24278 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 19 | RSCFD0CFDTMDF4_19 | 00000000 _H | 32 | FFD2427C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 20 | RSCFD0CFDTMID20 | 00000000 _H | 32 | FFD24280 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 20 | RSCFD0CFDTMPTR20 | 00000000 _H | 32 | FFD24284 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 20 | RSCFD0CFDTMFDCTR20 | 00000000 _H | 32 | FFD24288 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 20 | RSCFD0CFDTMDF0_20 | 00000000 _H | 32 | FFD2428C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 20 | RSCFD0CFDTMDF1_20 | 00000000 _H | 32 | FFD24290 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 20 | RSCFD0CFDTMDF2_20 | 00000000 _H | 32 | FFD24294 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 20 | RSCFD0CFDTMDF3_20 | 00000000 _H | 32 | FFD24298 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 20 | RSCFD0CFDTMDF4_20 | 00000000 _H | 32 | FFD2429C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 21 | RSCFD0CFDTMID21 | 00000000 _H | 32 | FFD242A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 21 | RSCFD0CFDTMPTR21 | 00000000 _H | 32 | FFD242A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 21 | RSCFD0CFDTMFDCTR21 | 00000000 _H | 32 | FFD242A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 21 | RSCFD0CFDTMDF0_21 | 00000000 _H | 32 | FFD242AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 21 | RSCFD0CFDTMDF1_21 | 00000000 _H | 32 | FFD242B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 21 | RSCFD0CFDTMDF2_21 | 00000000 _H | 32 | FFD242B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 21 | RSCFD0CFDTMDF3_21 | 00000000 _H | 32 | FFD242B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 21 | RSCFD0CFDTMDF4_21 | 00000000 _H | 32 | FFD242BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 22 | RSCFD0CFDTMID22 | 00000000 _H | 32 | FFD242C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 22 | RSCFD0CFDTMPTR22 | 00000000 _H | 32 | FFD242C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (71/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 22 | RSCFD0CFDTMFDCTR22 | 00000000 _H | 32 | FFD242C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 22 | RSCFD0CFDTMDF0_22 | 00000000 _H | 32 | FFD242CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 22 | RSCFD0CFDTMDF1_22 | 00000000 _H | 32 | FFD242D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 22 | RSCFD0CFDTMDF2_22 | 00000000 _H | 32 | FFD242D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 22 | RSCFD0CFDTMDF3_22 | 00000000 _H | 32 | FFD242D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 22 | RSCFD0CFDTMDF4_22 | 00000000 _H | 32 | FFD242DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 23 | RSCFD0CFDTMID23 | 00000000 _H | 32 | FFD242E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 23 | RSCFD0CFDTMPTR23 | 00000000 _H | 32 | FFD242E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 23 | RSCFD0CFDTMFDCTR23 | 00000000 _H | 32 | FFD242E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 23 | RSCFD0CFDTMDF0_23 | 00000000 _H | 32 | FFD242EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 23 | RSCFD0CFDTMDF1_23 | 00000000 _H | 32 | FFD242F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 23 | RSCFD0CFDTMDF2_23 | 00000000 _H | 32 | FFD242F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 23 | RSCFD0CFDTMDF3_23 | 00000000 _H | 32 | FFD242F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 23 | RSCFD0CFDTMDF4_23 | 00000000 _H | 32 | FFD242FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 24 | RSCFD0CFDTMID24 | 00000000 _H | 32 | FFD24300 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 24 | RSCFD0CFDTMPTR24 | 00000000 _H | 32 | FFD24304 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 24 | RSCFD0CFDTMFDCTR24 | 00000000 _H | 32 | FFD24308 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 24 | RSCFD0CFDTMDF0_24 | 00000000 _H | 32 | FFD2430C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 24 | RSCFD0CFDTMDF1_24 | 00000000 _H | 32 | FFD24310 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 24 | RSCFD0CFDTMDF2_24 | 00000000 _H | 32 | FFD24314 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 24 | RSCFD0CFDTMDF3_24 | 00000000 _H | 32 | FFD24318 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 24 | RSCFD0CFDTMDF4_24 | 00000000 _H | 32 | FFD2431C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 25 | RSCFD0CFDTMID25 | 00000000 _H | 32 | FFD24320 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 25 | RSCFD0CFDTMPTR25 | 00000000 _H | 32 | FFD24324 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 25 | RSCFD0CFDTMFDCTR25 | 00000000 _H | 32 | FFD24328 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 25 | RSCFD0CFDTMDF0_25 | 00000000 _H | 32 | FFD2432C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 25 | RSCFD0CFDTMDF1_25 | 00000000 _H | 32 | FFD24330 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 25 | RSCFD0CFDTMDF2_25 | 00000000 _H | 32 | FFD24334 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 25 | RSCFD0CFDTMDF3_25 | 00000000 _H | 32 | FFD24338 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 25 | RSCFD0CFDTMDF4_25 | 00000000 _H | 32 | FFD2433C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 26 | RSCFD0CFDTMID26 | 00000000 _H | 32 | FFD24340 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 26 | RSCFD0CFDTMPTR26 | 00000000 _H | 32 | FFD24344 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 26 | RSCFD0CFDTMFDCTR26 | 00000000 _H | 32 | FFD24348 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 26 | RSCFD0CFDTMDF0_26 | 00000000 _H | 32 | FFD2434C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 26 | RSCFD0CFDTMDF1_26 | 00000000 _H | 32 | FFD24350 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 26 | RSCFD0CFDTMDF2_26 | 00000000 _H | 32 | FFD24354 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 26 | RSCFD0CFDTMDF3_26 | 00000000 _H | 32 | FFD24358 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 26 | RSCFD0CFDTMDF4_26 | 00000000 _H | 32 | FFD2435C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 27 | RSCFD0CFDTMID27 | 00000000 _H | 32 | FFD24360 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 27 | RSCFD0CFDTMPTR27 | 00000000 _H | 32 | FFD24364 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 27 | RSCFD0CFDTMFDCTR27 | 00000000 _H | 32 | FFD24368 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 27 | RSCFD0CFDTMDF0_27 | 00000000 _H | 32 | FFD2436C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 27 | RSCFD0CFDTMDF1_27 | 00000000 _H | 32 | FFD24370 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 27 | RSCFD0CFDTMDF2_27 | 00000000 _H | 32 | FFD24374 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 27 | RSCFD0CFDTMDF3_27 | 00000000 _H | 32 | FFD24378 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 27 | RSCFD0CFDTMDF4_27 | 00000000 _H | 32 | FFD2437C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 28 | RSCFD0CFDTMID28 | 00000000 _H | 32 | FFD24380 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 28 | RSCFD0CFDTMPTR28 | 00000000 _H | 32 | FFD24384 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (72/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 28 | RSCFD0CFDTMFDCTR28 | 00000000 _H | 32 | FFD24388 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 28 | RSCFD0CFDTMDF0_28 | 00000000 _H | 32 | FFD2438C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 28 | RSCFD0CFDTMDF1_28 | 00000000 _H | 32 | FFD24390 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 28 | RSCFD0CFDTMDF2_28 | 00000000 _H | 32 | FFD24394 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 28 | RSCFD0CFDTMDF3_28 | 00000000 _H | 32 | FFD24398 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 28 | RSCFD0CFDTMDF4_28 | 00000000 _H | 32 | FFD2439C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 29 | RSCFD0CFDTMID29 | 00000000 _H | 32 | FFD243A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 29 | RSCFD0CFDTMPTR29 | 00000000 _H | 32 | FFD243A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 29 | RSCFD0CFDTMFDCTR29 | 00000000 _H | 32 | FFD243A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 29 | RSCFD0CFDTMDF0_29 | 00000000 _H | 32 | FFD243AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 29 | RSCFD0CFDTMDF1_29 | 00000000 _H | 32 | FFD243B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 29 | RSCFD0CFDTMDF2_29 | 00000000 _H | 32 | FFD243B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 29 | RSCFD0CFDTMDF3_29 | 00000000 _H | 32 | FFD243B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 29 | RSCFD0CFDTMDF4_29 | 00000000 _H | 32 | FFD243BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 30 | RSCFD0CFDTMID30 | 00000000 _H | 32 | FFD243C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 30 | RSCFD0CFDTMPTR30 | 00000000 _H | 32 | FFD243C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 30 | RSCFD0CFDTMFDCTR30 | 00000000 _H | 32 | FFD243C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 30 | RSCFD0CFDTMDF0_30 | 00000000 _H | 32 | FFD243CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 30 | RSCFD0CFDTMDF1_30 | 00000000 _H | 32 | FFD243D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 30 | RSCFD0CFDTMDF2_30 | 00000000 _H | 32 | FFD243D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 30 | RSCFD0CFDTMDF3_30 | 00000000 _H | 32 | FFD243D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 30 | RSCFD0CFDTMDF4_30 | 00000000 _H | 32 | FFD243DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 31 | RSCFD0CFDTMID31 | 00000000 _H | 32 | FFD243E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 31 | RSCFD0CFDTMPTR31 | 00000000 _H | 32 | FFD243E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 31 | RSCFD0CFDTMFDCTR31 | 00000000 _H | 32 | FFD243E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 31 | RSCFD0CFDTMDF0_31 | 00000000 _H | 32 | FFD243EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 31 | RSCFD0CFDTMDF1_31 | 00000000 _H | 32 | FFD243F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 31 | RSCFD0CFDTMDF2_31 | 00000000 _H | 32 | FFD243F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 31 | RSCFD0CFDTMDF3_31 | 00000000 _H | 32 | FFD243F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 31 | RSCFD0CFDTMDF4_31 | 00000000 _H | 32 | FFD243FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 32 | RSCFD0CFDTMID32 | 00000000 _H | 32 | FFD24400 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 32 | RSCFD0CFDTMPTR32 | 00000000 _H | 32 | FFD24404 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 32 | RSCFD0CFDTMFDCTR32 | 00000000 _H | 32 | FFD24408 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 32 | RSCFD0CFDTMDF0_32 | 00000000 _H | 32 | FFD2440C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 32 | RSCFD0CFDTMDF1_32 | 00000000 _H | 32 | FFD24410 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 32 | RSCFD0CFDTMDF2_32 | 00000000 _H | 32 | FFD24414 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 32 | RSCFD0CFDTMDF3_32 | 00000000 _H | 32 | FFD24418 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 32 | RSCFD0CFDTMDF4_32 | 00000000 _H | 32 | FFD2441C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 33 | RSCFD0CFDTMID33 | 00000000 _H | 32 | FFD24420 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 33 | RSCFD0CFDTMPTR33 | 00000000 _H | 32 | FFD24424 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 33 | RSCFD0CFDTMFDCTR33 | 00000000 _H | 32 | FFD24428 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 33 | RSCFD0CFDTMDF0_33 | 00000000 _H | 32 | FFD2442C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 33 | RSCFD0CFDTMDF1_33 | 00000000 _H | 32 | FFD24430 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 33 | RSCFD0CFDTMDF2_33 | 00000000 _H | 32 | FFD24434 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 33 | RSCFD0CFDTMDF3_33 | 00000000 _H | 32 | FFD24438 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 33 | RSCFD0CFDTMDF4_33 | 00000000 _H | 32 | FFD2443C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 34 | RSCFD0CFDTMID34 | 00000000 _H | 32 | FFD24440 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 34 | RSCFD0CFDTMPTR34 | 00000000 _H | 32 | FFD24444 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (73/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 34 | RSCFD0CFDTMFDCTR34 | 00000000 _H | 32 | FFD24448 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 34 | RSCFD0CFDTMDF0_34 | 00000000 _H | 32 | FFD2444C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 34 | RSCFD0CFDTMDF1_34 | 00000000 _H | 32 | FFD24450 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 34 | RSCFD0CFDTMDF2_34 | 00000000 _H | 32 | FFD24454 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 34 | RSCFD0CFDTMDF3_34 | 00000000 _H | 32 | FFD24458 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 34 | RSCFD0CFDTMDF4_34 | 00000000 _H | 32 | FFD2445C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 35 | RSCFD0CFDTMID35 | 00000000 _H | 32 | FFD24460 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 35 | RSCFD0CFDTMPTR35 | 00000000 _H | 32 | FFD24464 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 35 | RSCFD0CFDTMFDCTR35 | 00000000 _H | 32 | FFD24468 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 35 | RSCFD0CFDTMDF0_35 | 00000000 _H | 32 | FFD2446C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 35 | RSCFD0CFDTMDF1_35 | 00000000 _H | 32 | FFD24470 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 35 | RSCFD0CFDTMDF2_35 | 00000000 _H | 32 | FFD24474 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 35 | RSCFD0CFDTMDF3_35 | 00000000 _H | 32 | FFD24478 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 35 | RSCFD0CFDTMDF4_35 | 00000000 _H | 32 | FFD2447C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 36 | RSCFD0CFDTMID36 | 00000000 _H | 32 | FFD24480 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 36 | RSCFD0CFDTMPTR36 | 00000000 _H | 32 | FFD24484 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 36 | RSCFD0CFDTMFDCTR36 | 00000000 _H | 32 | FFD24488 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 36 | RSCFD0CFDTMDF0_36 | 00000000 _H | 32 | FFD2448C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 36 | RSCFD0CFDTMDF1_36 | 00000000 _H | 32 | FFD24490 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 36 | RSCFD0CFDTMDF2_36 | 00000000 _H | 32 | FFD24494 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 36 | RSCFD0CFDTMDF3_36 | 00000000 _H | 32 | FFD24498 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 36 | RSCFD0CFDTMDF4_36 | 00000000 _H | 32 | FFD2449C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 37 | RSCFD0CFDTMID37 | 00000000 _H | 32 | FFD244A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 37 | RSCFD0CFDTMPTR37 | 00000000 _H | 32 | FFD244A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 37 | RSCFD0CFDTMFDCTR37 | 00000000 _H | 32 | FFD244A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 37 | RSCFD0CFDTMDF0_37 | 00000000 _H | 32 | FFD244AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 37 | RSCFD0CFDTMDF1_37 | 00000000 _H | 32 | FFD244B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 37 | RSCFD0CFDTMDF2_37 | 00000000 _H | 32 | FFD244B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 37 | RSCFD0CFDTMDF3_37 | 00000000 _H | 32 | FFD244B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 37 | RSCFD0CFDTMDF4_37 | 00000000 _H | 32 | FFD244BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 38 | RSCFD0CFDTMID38 | 00000000 _H | 32 | FFD244C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 38 | RSCFD0CFDTMPTR38 | 00000000 _H | 32 | FFD244C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 38 | RSCFD0CFDTMFDCTR38 | 00000000 _H | 32 | FFD244C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 38 | RSCFD0CFDTMDF0_38 | 00000000 _H | 32 | FFD244CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 38 | RSCFD0CFDTMDF1_38 | 00000000 _H | 32 | FFD244D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 38 | RSCFD0CFDTMDF2_38 | 00000000 _H | 32 | FFD244D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 38 | RSCFD0CFDTMDF3_38 | 00000000 _H | 32 | FFD244D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 38 | RSCFD0CFDTMDF4_38 | 00000000 _H | 32 | FFD244DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 39 | RSCFD0CFDTMID39 | 00000000 _H | 32 | FFD244E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 39 | RSCFD0CFDTMPTR39 | 00000000 _H | 32 | FFD244E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 39 | RSCFD0CFDTMFDCTR39 | 00000000 _H | 32 | FFD244E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 39 | RSCFD0CFDTMDF0_39 | 00000000 _H | 32 | FFD244EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 39 | RSCFD0CFDTMDF1_39 | 00000000 _H | 32 | FFD244F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 39 | RSCFD0CFDTMDF2_39 | 00000000 _H | 32 | FFD244F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 39 | RSCFD0CFDTMDF3_39 | 00000000 _H | 32 | FFD244F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 39 | RSCFD0CFDTMDF4_39 | 00000000 _H | 32 | FFD244FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 40 | RSCFD0CFDTMID40 | 00000000 _H | 32 | FFD24500 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 40 | RSCFD0CFDTMPTR40 | 00000000 _H | 32 | FFD24504 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (74/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 40 | RSCFD0CFDTMFDCTR40 | 00000000 _H | 32 | FFD24508 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 40 | RSCFD0CFDTMDF0_40 | 00000000 _H | 32 | FFD2450C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 40 | RSCFD0CFDTMDF1_40 | 00000000 _H | 32 | FFD24510 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 40 | RSCFD0CFDTMDF2_40 | 00000000 _H | 32 | FFD24514 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 40 | RSCFD0CFDTMDF3_40 | 00000000 _H | 32 | FFD24518 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 40 | RSCFD0CFDTMDF4_40 | 00000000 _H | 32 | FFD2451C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 41 | RSCFD0CFDTMID41 | 00000000 _H | 32 | FFD24520 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 41 | RSCFD0CFDTMPTR41 | 00000000 _H | 32 | FFD24524 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 41 | RSCFD0CFDTMFDCTR41 | 00000000 _H | 32 | FFD24528 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 41 | RSCFD0CFDTMDF0_41 | 00000000 _H | 32 | FFD2452C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 41 | RSCFD0CFDTMDF1_41 | 00000000 _H | 32 | FFD24530 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 41 | RSCFD0CFDTMDF2_41 | 00000000 _H | 32 | FFD24534 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 41 | RSCFD0CFDTMDF3_41 | 00000000 _H | 32 | FFD24538 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 41 | RSCFD0CFDTMDF4_41 | 00000000 _H | 32 | FFD2453C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 42 | RSCFD0CFDTMID42 | 00000000 _H | 32 | FFD24540 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 42 | RSCFD0CFDTMPTR42 | 00000000 _H | 32 | FFD24544 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 42 | RSCFD0CFDTMFDCTR42 | 00000000 _H | 32 | FFD24548 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 42 | RSCFD0CFDTMDF0_42 | 00000000 _H | 32 | FFD2454C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 42 | RSCFD0CFDTMDF1_42 | 00000000 _H | 32 | FFD24550 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 42 | RSCFD0CFDTMDF2_42 | 00000000 _H | 32 | FFD24554 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 42 | RSCFD0CFDTMDF3_42 | 00000000 _H | 32 | FFD24558 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 42 | RSCFD0CFDTMDF4_42 | 00000000 _H | 32 | FFD2455C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 43 | RSCFD0CFDTMID43 | 00000000 _H | 32 | FFD24560 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 43 | RSCFD0CFDTMPTR43 | 00000000 _H | 32 | FFD24564 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 43 | RSCFD0CFDTMFDCTR43 | 00000000 _H | 32 | FFD24568 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 43 | RSCFD0CFDTMDF0_43 | 00000000 _H | 32 | FFD2456C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 43 | RSCFD0CFDTMDF1_43 | 00000000 _H | 32 | FFD24570 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 43 | RSCFD0CFDTMDF2_43 | 00000000 _H | 32 | FFD24574 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 43 | RSCFD0CFDTMDF3_43 | 00000000 _H | 32 | FFD24578 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 43 | RSCFD0CFDTMDF4_43 | 00000000 _H | 32 | FFD2457C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 44 | RSCFD0CFDTMID44 | 00000000 _H | 32 | FFD24580 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 44 | RSCFD0CFDTMPTR44 | 00000000 _H | 32 | FFD24584 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 44 | RSCFD0CFDTMFDCTR44 | 00000000 _H | 32 | FFD24588 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 44 | RSCFD0CFDTMDF0_44 | 00000000 _H | 32 | FFD2458C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 44 | RSCFD0CFDTMDF1_44 | 00000000 _H | 32 | FFD24590 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 44 | RSCFD0CFDTMDF2_44 | 00000000 _H | 32 | FFD24594 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 44 | RSCFD0CFDTMDF3_44 | 00000000 _H | 32 | FFD24598 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 44 | RSCFD0CFDTMDF4_44 | 00000000 _H | 32 | FFD2459C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 45 | RSCFD0CFDTMID45 | 00000000 _H | 32 | FFD245A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 45 | RSCFD0CFDTMPTR45 | 00000000 _H | 32 | FFD245A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 45 | RSCFD0CFDTMFDCTR45 | 00000000 _H | 32 | FFD245A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 45 | RSCFD0CFDTMDF0_45 | 00000000 _H | 32 | FFD245AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 45 | RSCFD0CFDTMDF1_45 | 00000000 _H | 32 | FFD245B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 45 | RSCFD0CFDTMDF2_45 | 00000000 _H | 32 | FFD245B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 45 | RSCFD0CFDTMDF3_45 | 00000000 _H | 32 | FFD245B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 45 | RSCFD0CFDTMDF4_45 | 00000000 _H | 32 | FFD245BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 46 | RSCFD0CFDTMID46 | 00000000 _H | 32 | FFD245C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 46 | RSCFD0CFDTMPTR46 | 00000000 _H | 32 | FFD245C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (75/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | Transmit buffer CAN FD configuration register 46 | RSCFD0CFDTMFDCTR46 | 00000000 _H | 32 | FFD245C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 46 | RSCFD0CFDTMDF0_46 | 00000000 _H | 32 | FFD245CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 46 | RSCFD0CFDTMDF1_46 | 00000000 _H | 32 | FFD245D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 46 | RSCFD0CFDTMDF2_46 | 00000000 _H | 32 | FFD245D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 46 | RSCFD0CFDTMDF3_46 | 00000000 _H | 32 | FFD245D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 46 | RSCFD0CFDTMDF4_46 | 00000000 _H | 32 | FFD245DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer ID register 47 | RSCFD0CFDTMID47 | 00000000 _H | 32 | FFD245E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer pointer register 47 | RSCFD0CFDTMPTR47 | 00000000 _H | 32 | FFD245E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit buffer CAN FD configuration register 47 | RSCFD0CFDTMFDCTR47 | 00000000 _H | 32 | FFD245E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 0 Register 47 | RSCFD0CFDTMDF0_47 | 00000000 _H | 32 | FFD245EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 1 Register 47 | RSCFD0CFDTMDF1_47 | 00000000 _H | 32 | FFD245F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 2 Register 47 | RSCFD0CFDTMDF2_47 | 00000000 _H | 32 | FFD245F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 3 Register 47 | RSCFD0CFDTMDF3_47 | 00000000 _H | 32 | FFD245F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit Buffer Data Field 4 Register 47 | RSCFD0CFDTMDF4_47 | 00000000 _H | 32 | FFD245FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history access register 0 | RSCFD0CFDTHLACC0 | 00000000 _H | 32 | FFD26000 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history access register 1 | RSCFD0CFDTHLACC1 | 00000000 _H | 32 | FFD26004 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | Transmit history access register 2 | RSCFD0CFDTHLACC2 | 00000000 _H | 32 | FFD26008 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 0 | RSCFD0CFDRPGACC0 | 00000000 _H | 32 | FFD26400 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 1 | RSCFD0CFDRPGACC1 | 00000000 _H | 32 | FFD26404 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 2 | RSCFD0CFDRPGACC2 | 00000000 _H | 32 | FFD26408 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 3 | RSCFD0CFDRPGACC3 | 00000000 _H | 32 | FFD2640C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 4 | RSCFD0CFDRPGACC4 | 00000000 _H | 32 | FFD26410 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 5 | RSCFD0CFDRPGACC5 | 00000000 _H | 32 | FFD26414 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 6 | RSCFD0CFDRPGACC6 | 00000000 _H | 32 | FFD26418 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 7 | RSCFD0CFDRPGACC7 | 00000000 _H | 32 | FFD2641C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 8 | RSCFD0CFDRPGACC8 | 00000000 _H | 32 | FFD26420 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 9 | RSCFD0CFDRPGACC9 | 00000000 _H | 32 | FFD26424 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 10 | RSCFD0CFDRPGACC10 | 00000000 _H | 32 | FFD26428 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 11 | RSCFD0CFDRPGACC11 | 00000000 _H | 32 | FFD2642C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 12 | RSCFD0CFDRPGACC12 | 00000000 _H | 32 | FFD26430 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 13 | RSCFD0CFDRPGACC13 | 00000000 _H | 32 | FFD26434 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 14 | RSCFD0CFDRPGACC14 | 00000000 _H | 32 | FFD26438 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 15 | RSCFD0CFDRPGACC15 | 00000000 _H | 32 | FFD2643C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 16 | RSCFD0CFDRPGACC16 | 00000000 _H | 32 | FFD26440 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 17 | RSCFD0CFDRPGACC17 | 00000000 _H | 32 | FFD26444 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 18 | RSCFD0CFDRPGACC18 | 00000000 _H | 32 | FFD26448 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 19 | RSCFD0CFDRPGACC19 | 00000000 _H | 32 | FFD2644C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 20 | RSCFD0CFDRPGACC20 | 00000000 _H | 32 | FFD26450 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 21 | RSCFD0CFDRPGACC21 | 00000000 _H | 32 | FFD26454 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 22 | RSCFD0CFDRPGACC22 | 00000000 _H | 32 | FFD26458 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 23 | RSCFD0CFDRPGACC23 | 00000000 _H | 32 | FFD2645C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 24 | RSCFD0CFDRPGACC24 | 00000000 _H | 32 | FFD26460 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 25 | RSCFD0CFDRPGACC25 | 00000000 _H | 32 | FFD26464 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 26 | RSCFD0CFDRPGACC26 | 00000000 _H | 32 | FFD26468 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 27 | RSCFD0CFDRPGACC27 | 00000000 _H | 32 | FFD2646C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 28 | RSCFD0CFDRPGACC28 | 00000000 _H | 32 | FFD26470 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 29 | RSCFD0CFDRPGACC29 | 00000000 _H | 32 | FFD26474 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 30 | RSCFD0CFDRPGACC30 | 00000000 _H | 32 | FFD26478 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (76/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---------------------------------------|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSCFD0 | RAM test page access register 31 | RSCFD0CFDRPGACC31 | 00000000 _H | 32 | FFD2647C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 32 | RSCFD0CFDRPGACC32 | 00000000 _H | 32 | FFD26480 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 33 | RSCFD0CFDRPGACC33 | 00000000 _H | 32 | FFD26484 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 34 | RSCFD0CFDRPGACC34 | 00000000 _H | 32 | FFD26488 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 35 | RSCFD0CFDRPGACC35 | 00000000 _H | 32 | FFD2648C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 36 | RSCFD0CFDRPGACC36 | 00000000 _H | 32 | FFD26490 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 37 | RSCFD0CFDRPGACC37 | 00000000 _H | 32 | FFD26494 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 38 | RSCFD0CFDRPGACC38 | 00000000 _H | 32 | FFD26498 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 39 | RSCFD0CFDRPGACC39 | 00000000 _H | 32 | FFD2649C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 40 | RSCFD0CFDRPGACC40 | 00000000 _H | 32 | FFD264A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 41 | RSCFD0CFDRPGACC41 | 00000000 _H | 32 | FFD264A4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 42 | RSCFD0CFDRPGACC42 | 00000000 _H | 32 | FFD264A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 43 | RSCFD0CFDRPGACC43 | 00000000 _H | 32 | FFD264AC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 44 | RSCFD0CFDRPGACC44 | 00000000 _H | 32 | FFD264B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 45 | RSCFD0CFDRPGACC45 | 00000000 _H | 32 | FFD264B4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 46 | RSCFD0CFDRPGACC46 | 00000000 _H | 32 | FFD264B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 47 | RSCFD0CFDRPGACC47 | 00000000 _H | 32 | FFD264BC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 48 | RSCFD0CFDRPGACC48 | 00000000 _H | 32 | FFD264C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 49 | RSCFD0CFDRPGACC49 | 00000000 _H | 32 | FFD264C4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 50 | RSCFD0CFDRPGACC50 | 00000000 _H | 32 | FFD264C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 51 | RSCFD0CFDRPGACC51 | 00000000 _H | 32 | FFD264CC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 52 | RSCFD0CFDRPGACC52 | 00000000 _H | 32 | FFD264D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 53 | RSCFD0CFDRPGACC53 | 00000000 _H | 32 | FFD264D4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 54 | RSCFD0CFDRPGACC54 | 00000000 _H | 32 | FFD264D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 55 | RSCFD0CFDRPGACC55 | 00000000 _H | 32 | FFD264DC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 56 | RSCFD0CFDRPGACC56 | 00000000 _H | 32 | FFD264E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 57 | RSCFD0CFDRPGACC57 | 00000000 _H | 32 | FFD264E4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 58 | RSCFD0CFDRPGACC58 | 00000000 _H | 32 | FFD264E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 59 | RSCFD0CFDRPGACC59 | 00000000 _H | 32 | FFD264EC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 60 | RSCFD0CFDRPGACC60 | 00000000 _H | 32 | FFD264F0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 61 | RSCFD0CFDRPGACC61 | 00000000 _H | 32 | FFD264F4 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 62 | RSCFD0CFDRPGACC62 | 00000000 _H | 32 | FFD264F8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | RAM test page access register 63 | RSCFD0CFDRPGACC63 | 00000000 _H | 32 | FFD264FC _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCAN0 | CAN FD mode read register | RSCAN0CANFMDR | 00000000 _H | 32 | FFD28000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSCFD0 | CAN FD mode read register | RSCFD0CANFMDR | 00000000 _H | 32 | FFD28000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA0 | CRC0 input register | DCRA0CIN | 00000000 _H | 32 | FFD50000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA0 | CRC0 data register | DCRA0COUT | 00000000 _H | 32 | FFD50004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA0 | CRC0 control register | DCRA0CTL | 00 _H | 8 | FFD50020 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA1 | CRC1 input register | DCRA1CIN | 00000000 _H | 32 | FFD51000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA1 | CRC1 data register | DCRA1COUT | 00000000 _H | 32 | FFD51004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA1 | CRC1 control register | DCRA1CTL | 00 _H | 8 | FFD51020 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA2 | CRC2 input register | DCRA2CIN | 00000000 _H | 32 | FFD52000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA2 | CRC2 data register | DCRA2COUT | 00000000 _H | 32 | FFD52004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA2 | CRC2 control register | DCRA2CTL | 00 _H | 8 | FFD52020 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA3 | CRC3 input register | DCRA3CIN | 00000000 _H | 32 | FFD53000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA3 | CRC3 data register | DCRA3COUT | 00000000 _H | 32 | FFD53004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DCRA3 | CRC3 control register | DCRA3CTL | 00 _H | 8 | FFD53020 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECMM | ECM master error set trigger register | ECMMESET | 00 _H | 8 | FFD60000 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (77/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ECMM | ECM master error clear trigger register | ECMMECLR | 00 _H | 8 | FFD60004 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| ECMM | ECM master error source status register 0 | ECMMESSTR0 | 00000000 _H | 32 | FFD60008 _H | 2 | 32 | √ (Except Debug Initiated Reset.) | — | — | — | √ | √ |
| ECMM | ECM master error source status register 1 | ECMMESSTR1 | 00000000 _H | 32 | FFD6000C _H | 2 | 32 | √ (Except Debug Initiated Reset.) | — | — | — | √ | √ |
| ECMM | ECM master error source status register 2 | ECMMESSTR2 | 00000000 _H | 32 | FFD60010 _H | 2 | 32 | √ (Except Debug Initiated Reset. The bit31 isn't initialized.) | — | — | — | √ | √ |
| ECMM | ECM master protection command register | ECMPCMD0 | Undefined | 32 | FFD60014 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECMC | ECM checker error set trigger register | ECMCESET | 00 _H | 8 | FFD61000 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| ECMC | ECM checker error clear trigger register | ECMCECLR | 00 _H | 8 | FFD61004 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| ECMC | ECM checker error source status register 0 | ECMCESSTR0 | 00000000 _H | 32 | FFD61008 _H | 2 | 32 | √ (Except Debug Initiated Reset.) | — | — | — | √ | √ |
| ECMC | ECM checker error source status register 1 | ECMCESSTR1 | 00000000 _H | 32 | FFD6100C _H | 2 | 32 | √ (Except Debug Initiated Reset.) | — | — | — | √ | √ |
| ECMC | ECM checker error source status register 2 | ECMCESSTR2 | 00000000 _H | 32 | FFD61010 _H | 2 | 32 | √ (Except Debug Initiated Reset. The bit31 isn't initialized.) | — | — | — | √ | √ |
| ECMC | ECM checker protection command register | ECMPCMD0 | Undefined | 32 | FFD61014 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM error pulse configuration register | ECMEPCFG | 00 _H | 8 | FFD62000 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| ECM | ECM maskable interrupt configuration register 0 | ECMMICFG0 | 00000000 _H | 32 | FFD62004 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM maskable interrupt configuration register 1 | ECMMICFG1 | 00000000 _H | 32 | FFD62008 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM maskable interrupt configuration register 2 | ECMMICFG2 | 00000000 _H | 32 | FFD6200C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM non-maskable interrupt configuration register 0 | ECMNMICFG0 | 00000000 _H | 32 | FFD62010 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM non-maskable interrupt configuration register 1 | ECMNMICFG1 | 00000000 _H | 32 | FFD62014 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM non-maskable interrupt configuration register 2 | ECMNMICFG2 | 00000000 _H | 32 | FFD62018 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM internal reset configuration register 0 | ECMIRCFG0 | 00000001 _H | 32 | FFD6201C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM internal reset configuration register 1 | ECMIRCFG1 | 00000000 _H | 32 | FFD62020 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM internal reset configuration register 2 | ECMIRCFG2 | 00000000 _H | 32 | FFD62024 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM error mask register 0 | ECMEMK0 | 00000000 _H | 32 | FFD62028 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM error mask register 1 | ECMEMK1 | 00000000 _H | 32 | FFD6202C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM error mask register 2 | ECMEMK2 | 00000000 _H | 32 | FFD62030 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM error source status clear register 0 | ECMESSTC0 | 00000000 _H | 32 | FFD62034 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM error source status clear register 1 | ECMESSTC1 | 00000000 _H | 32 | FFD62038 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM error source status clear register 2 | ECMESSTC2 | 00000000 _H | 32 | FFD6203C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM protection command register | ECMPCMD1 | Undefined | 32 | FFD62040 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM protection status register | ECMPS | 00 _H | 8 | FFD62044 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| ECM | ECM pseudo error trigger register 0 | ECMPE0 | 00000000 _H | 32 | FFD62048 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM pseudo error trigger register 1 | ECMPE1 | 00000000 _H | 32 | FFD6204C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM pseudo error trigger register 2 | ECMPE2 | 00000000 _H | 32 | FFD62050 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| ECM | ECM delay timer control register | ECMDTMCTL | 00 _H | 8 | FFD62054 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| ECM | ECM delay timer register | ECMDTMR | 0000 _H | 16 | FFD62058 _H | 2 | 16 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (78/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|------------|--|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ECM | ECM delay timer compare register | ECMDTMCMP | 00000000 _H | 32 | FFD6205C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM delay timer configuration register 0 | ECMDTMCFG0 | 00000000 _H | 32 | FFD62060 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM delay timer configuration register 1 | ECMDTMCFG1 | 00000000 _H | 32 | FFD62064 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM delay timer configuration register 2 | ECMDTMCFG2 | 00000000 _H | 32 | FFD62068 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM delay timer configuration register 3 | ECMDTMCFG3 | 00000000 _H | 32 | FFD6206C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM delay timer configuration register 4 | ECMDTMCFG4 | 00000000 _H | 32 | FFD62070 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM delay timer configuration register 5 | ECMDTMCFG5 | 00000000 _H | 32 | FFD62074 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM error output clear invalidation configuration register | ECMEOCCFG | 00000000 _H | 32 | FFD62078 _H | 2 | 32 | ✓ | ✓ | ✓ | — | ✓ | ✓ |
| ECM | ECM pseudo error mask register | ECMPEM | 00000000 _H | 32 | FFD6207C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECM | ECM error pulse control register | ECMEPCTL | 00 _H | 8 | FFD63000 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SYS | FEINT Factor Register | FEINTF | 00000000 _H | 32 | FFD67000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SYS | FEINT Factor Clear Register | FEINTFC | 00000000 _H | 32 | FFD67008 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM3 | OSTM3 compare register | OSTM3CMP | 00000000 _H | 32 | FFD70000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM3 | OSTM3 counter register | OSTM3CNT | FFFFFFFF _H | 32 | FFD70004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM3 | OSTM3 count enable status register | OSTM3TE | 00 _H | 8 | FFD70010 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM3 | OSTM3 count start trigger register | OSTM3TS | 00 _H | 8 | FFD70014 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM3 | OSTM3 count stop trigger register | OSTM3TT | 00 _H | 8 | FFD70018 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM3 | OSTM3 control register | OSTM3CTL | 00 _H | 8 | FFD70020 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM4 | OSTM4 compare register | OSTM4CMP | 00000000 _H | 32 | FFD70040 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM4 | OSTM4 counter register | OSTM4CNT | FFFFFFFF _H | 32 | FFD70044 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM4 | OSTM4 count enable status register | OSTM4TE | 00 _H | 8 | FFD70050 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM4 | OSTM4 count start trigger register | OSTM4TS | 00 _H | 8 | FFD70054 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM4 | OSTM4 count stop trigger register | OSTM4TT | 00 _H | 8 | FFD70058 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM4 | OSTM4 control register | OSTM4CTL | 00 _H | 8 | FFD70060 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM5 | OSTM5 compare register | OSTM5CMP | 00000000 _H | 32 | FFD70080 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM5 | OSTM5 counter register | OSTM5CNT | FFFFFFFF _H | 32 | FFD70084 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM5 | OSTM5 count enable status register | OSTM5TE | 00 _H | 8 | FFD70090 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM5 | OSTM5 count start trigger register | OSTM5TS | 00 _H | 8 | FFD70094 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM5 | OSTM5 count stop trigger register | OSTM5TT | 00 _H | 8 | FFD70098 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM5 | OSTM5 control register | OSTM5CTL | 00 _H | 8 | FFD700A0 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM6 | OSTM6 compare register | OSTM6CMP | 00000000 _H | 32 | FFD700C0 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM6 | OSTM6 counter register | OSTM6CNT | FFFFFFFF _H | 32 | FFD700C4 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM6 | OSTM6 count enable status register | OSTM6TE | 00 _H | 8 | FFD700D0 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM6 | OSTM6 count start trigger register | OSTM6TS | 00 _H | 8 | FFD700D4 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM6 | OSTM6 count stop trigger register | OSTM6TT | 00 _H | 8 | FFD700D8 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM6 | OSTM6 control register | OSTM6CTL | 00 _H | 8 | FFD700E0 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM7 | OSTM7 compare register | OSTM7CMP | 00000000 _H | 32 | FFD70100 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM7 | OSTM7 counter register | OSTM7CNT | FFFFFFFF _H | 32 | FFD70104 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM7 | OSTM7 count enable status register | OSTM7TE | 00 _H | 8 | FFD70110 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM7 | OSTM7 count start trigger register | OSTM7TS | 00 _H | 8 | FFD70114 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM7 | OSTM7 count stop trigger register | OSTM7TT | 00 _H | 8 | FFD70118 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM7 | OSTM7 control register | OSTM7CTL | 00 _H | 8 | FFD70120 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| WDTA0 | WDTA enable register | WDTA0WDTE | The initial value depends on the start-up options OPWDRUN and OPWDVAC. | 8 | FFD74000 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (79/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|--|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| WDTA0 | WDTA enable VAC register | WDTA0EVAC | The initial value depends on the start-up options OPWDRUN and OPWDVAC. | 8 | FFD74004 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| WDTA0 | WDTA reference value register | WDTA0REF | 00 _H | 8 | FFD74008 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| WDTA0 | WDTA mode register | WDTA0MD | The initial value depends on the start-up options OPWDOVF [2:0]. | 8 | FFD7400C _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 control register 0 | CSIH0CTL0 | 00 _H | 8 | FFD80000 _H | 2 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 status register 0 | CSIH0STR0 | 00000010 _H | 32 | FFD80004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 status clear register 0 | CSIH0STCR0 | 0000 _H | 16 | FFD80008 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 control register 1 | CSIH0CTL1 | 00000000 _H | 32 | FFD80010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 control register 2 | CSIH0CTL2 | E000 _H | 16 | FFD80014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 memory control register 1 | CSIH0MCTL1 | 00000000 _H | 32 | FFD81000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 memory control register 2 | CSIH0MCTL2 | 00000000 _H | 32 | FFD81004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 transmit data register 0 for word access | CSIH0TX0W | X0XXXXXX _H | 32 | FFD81008 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 transmit data register 0 for half word access | CSIH0TX0H | Undefined | 16 | FFD8100C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 receive data register 0 for word access | CSIH0RX0W | 0XXXXXXX _H | 32 | FFD81010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 receive data register 0 for half word access | CSIH0RX0H | Undefined | 16 | FFD81014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 memory read/write pointer register 0 | CSIH0MRWP0 | 00000000 _H | 32 | FFD81018 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 memory control register 0 | CSIH0MCTL0 | 001F _H | 16 | FFD81040 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 0 | CSIH0CFG0 | 00000000 _H | 32 | FFD81044 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 1 | CSIH0CFG1 | 00000000 _H | 32 | FFD81048 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 2 | CSIH0CFG2 | 00000000 _H | 32 | FFD8104C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 3 | CSIH0CFG3 | 00000000 _H | 32 | FFD81050 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 4 | CSIH0CFG4 | 00000000 _H | 32 | FFD81054 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 5 | CSIH0CFG5 | 00000000 _H | 32 | FFD81058 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 6 | CSIH0CFG6 | 00000000 _H | 32 | FFD8105C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 configuration register 7 | CSIH0CFG7 | 00000000 _H | 32 | FFD81060 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 baud rate setting register 0 | CSIH0BRS0 | 0000 _H | 16 | FFD81068 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 baud rate setting register 1 | CSIH0BRS1 | 0000 _H | 16 | FFD8106C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 baud rate setting register 2 | CSIH0BRS2 | 0000 _H | 16 | FFD81070 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH0 | CSIH0 baud rate setting register 3 | CSIH0BRS3 | 0000 _H | 16 | FFD81074 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 control register 0 | CSIH1CTL0 | 00 _H | 8 | FFD82000 _H | 2 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 status register 0 | CSIH1STR0 | 00000010 _H | 32 | FFD82004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 status clear register 0 | CSIH1STCR0 | 0000 _H | 16 | FFD82008 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 control register 1 | CSIH1CTL1 | 00000000 _H | 32 | FFD82010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 control register 2 | CSIH1CTL2 | E000 _H | 16 | FFD82014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 memory control register 1 | CSIH1MCTL1 | 00000000 _H | 32 | FFD83000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 memory control register 2 | CSIH1MCTL2 | 00000000 _H | 32 | FFD83004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 transmit data register 0 for word access | CSIH1TX0W | X0XXXXXX _H | 32 | FFD83008 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 transmit data register 0 for half word access | CSIH1TX0H | Undefined | 16 | FFD8300C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 receive data register 0 for word access | CSIH1RX0W | 0XXXXXXX _H | 32 | FFD83010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 receive data register 0 for half word access | CSIH1RX0H | Undefined | 16 | FFD83014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 memory read/write pointer register 0 | CSIH1MRWP0 | 00000000 _H | 32 | FFD83018 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 memory control register 0 | CSIH1MCTL0 | 001F _H | 16 | FFD83040 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 configuration register 0 | CSIH1CFG0 | 00000000 _H | 32 | FFD83044 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 configuration register 1 | CSIH1CFG1 | 00000000 _H | 32 | FFD83048 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (80/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| CSIH1 | CSIH1 configuration register 2 | CSIH1CFG2 | 00000000 _H | 32 | FFD8304C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 configuration register 3 | CSIH1CFG3 | 00000000 _H | 32 | FFD83050 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 configuration register 4 | CSIH1CFG4 | 00000000 _H | 32 | FFD83054 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 configuration register 5 | CSIH1CFG5 | 00000000 _H | 32 | FFD83058 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 configuration register 6 | CSIH1CFG6 | 00000000 _H | 32 | FFD8305C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 configuration register 7 | CSIH1CFG7 | 00000000 _H | 32 | FFD83060 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 baud rate setting register 0 | CSIH1BRS0 | 0000 _H | 16 | FFD83068 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 baud rate setting register 1 | CSIH1BRS1 | 0000 _H | 16 | FFD8306C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 baud rate setting register 2 | CSIH1BRS2 | 0000 _H | 16 | FFD83070 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH1 | CSIH1 baud rate setting register 3 | CSIH1BRS3 | 0000 _H | 16 | FFD83074 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 control register 0 | CSIH2CTL0 | 00 _H | 8 | FFD84000 _H | 2 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 status register 0 | CSIH2STR0 | 00000010 _H | 32 | FFD84004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 status clear register 0 | CSIH2STCR0 | 0000 _H | 16 | FFD84008 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 control register 1 | CSIH2CTL1 | 00000000 _H | 32 | FFD84010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 control register 2 | CSIH2CTL2 | E000 _H | 16 | FFD84014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 memory control register 1 | CSIH2MCTL1 | 00000000 _H | 32 | FFD85000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 memory control register 2 | CSIH2MCTL2 | 00000000 _H | 32 | FFD85004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 transmit data register 0 for word access | CSIH2TX0W | X0XXXXXX _H | 32 | FFD85008 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 transmit data register 0 for half word access | CSIH2TX0H | Undefined | 16 | FFD8500C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 receive data register 0 for word access | CSIH2RX0W | 0XXXXXXX _H | 32 | FFD85010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 receive data register 0 for half word access | CSIH2RX0H | Undefined | 16 | FFD85014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 memory read/write pointer register 0 | CSIH2MRWP0 | 00000000 _H | 32 | FFD85018 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 memory control register 0 | CSIH2MCTL0 | 001F _H | 16 | FFD85040 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 0 | CSIH2CFG0 | 00000000 _H | 32 | FFD85044 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 1 | CSIH2CFG1 | 00000000 _H | 32 | FFD85048 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 2 | CSIH2CFG2 | 00000000 _H | 32 | FFD8504C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 3 | CSIH2CFG3 | 00000000 _H | 32 | FFD85050 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 4 | CSIH2CFG4 | 00000000 _H | 32 | FFD85054 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 5 | CSIH2CFG5 | 00000000 _H | 32 | FFD85058 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 6 | CSIH2CFG6 | 00000000 _H | 32 | FFD8505C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 configuration register 7 | CSIH2CFG7 | 00000000 _H | 32 | FFD85060 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 baud rate setting register 0 | CSIH2BRS0 | 0000 _H | 16 | FFD85068 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 baud rate setting register 1 | CSIH2BRS1 | 0000 _H | 16 | FFD8506C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 baud rate setting register 2 | CSIH2BRS2 | 0000 _H | 16 | FFD85070 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH2 | CSIH2 baud rate setting register 3 | CSIH2BRS3 | 0000 _H | 16 | FFD85074 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 control register 0 | CSIH3CTL0 | 00 _H | 8 | FFD86000 _H | 2 | 1, 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 status register 0 | CSIH3STR0 | 00000010 _H | 32 | FFD86004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 status clear register 0 | CSIH3STCR0 | 0000 _H | 16 | FFD86008 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 control register 1 | CSIH3CTL1 | 00000000 _H | 32 | FFD86010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 control register 2 | CSIH3CTL2 | E000 _H | 16 | FFD86014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 memory control register 1 | CSIH3MCTL1 | 00000000 _H | 32 | FFD87000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 memory control register 2 | CSIH3MCTL2 | 00000000 _H | 32 | FFD87004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 transmit data register 0 for word access | CSIH3TX0W | X0XXXXXX _H | 32 | FFD87008 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 transmit data register 0 for half word access | CSIH3TX0H | Undefined | 16 | FFD8700C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 receive data register 0 for word access | CSIH3RX0W | 0XXXXXXX _H | 32 | FFD87010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 receive data register 0 for half word access | CSIH3RX0H | Undefined | 16 | FFD87014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 memory read/write pointer register 0 | CSIH3MRWP0 | 00000000 _H | 32 | FFD87018 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (81/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| CSIH3 | CSIH3 memory control register 0 | CSIH3MCTL0 | 001F _H | 16 | FFD87040 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 0 | CSIH3CFG0 | 00000000 _H | 32 | FFD87044 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 1 | CSIH3CFG1 | 00000000 _H | 32 | FFD87048 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 2 | CSIH3CFG2 | 00000000 _H | 32 | FFD8704C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 3 | CSIH3CFG3 | 00000000 _H | 32 | FFD87050 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 4 | CSIH3CFG4 | 00000000 _H | 32 | FFD87054 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 5 | CSIH3CFG5 | 00000000 _H | 32 | FFD87058 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 6 | CSIH3CFG6 | 00000000 _H | 32 | FFD8705C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 configuration register 7 | CSIH3CFG7 | 00000000 _H | 32 | FFD87060 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 baud rate setting register 0 | CSIH3BRS0 | 0000 _H | 16 | FFD87068 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 baud rate setting register 1 | CSIH3BRS1 | 0000 _H | 16 | FFD8706C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 baud rate setting register 2 | CSIH3BRS2 | 0000 _H | 16 | FFD87070 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH3 | CSIH3 baud rate setting register 3 | CSIH3BRS3 | 0000 _H | 16 | FFD87074 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 control register 0 | CSIG0CTL0 | 00 _H | 8 | FFD8A000 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 status register 0 | CSIG0STR0 | 00000010 _H | 32 | FFD8A004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 status clear register 0 | CSIG0STCR0 | 0000 _H | 16 | FFD8A008 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 control register 1 | CSIG0CTL1 | 00000000 _H | 32 | FFD8A010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 control register 2 | CSIG0CTL2 | E000 _H | 16 | FFD8A014 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 Rx-only mode control register 0 | CSIG0BCTL0 | 01 _H | 8 | FFD8B000 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIGn transmission register 0 for word access | CSIG0TX0W | 00000000 _H | 32 | FFD8B004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIGn transmission register 0 for half word access | CSIG0TX0H | 0000 _H | 16 | FFD8B008 _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 reception register 0 | CSIG0RX0 | 0000 _H | 16 | FFD8B00C _H | 2 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIG0 | CSIG0 configuration register 0 | CSIG0CFG0 | 00000000 _H | 32 | FFD8B010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSIH | CSIH DMA select register | SELCSIHDMA | 00 _H | 8 | FFD8E000 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Simultaneous start trigger control register | PIC1ASST | 00 _H | 8 | FFDD0004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Simultaneous start control register 0 | PIC1ASSER0 | 0000 _H | 16 | FFDD0010 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Simultaneous start control register 1 | PIC1ASSER1 | 0000 _H | 16 | FFDD0014 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Simultaneous start control register 2 | PIC1ASSER2 | 0000 _H | 16 | FFDD0018 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Simultaneous start control register 3 | PIC1ASSER3 | 0000 _H | 16 | FFDD001C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | RS flip-flop circuit initialization register 00 | PIC1AINI00 | 00 _H | 8 | FFDD0020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | DT initialization register 01 | PIC1AINI01 | 00 _H | 8 | FFDD0024 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | RS flip-flop circuit initialization register 10 | PIC1AINI10 | 00 _H | 8 | FFDD002C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | DT initialization register 11 | PIC1AINI11 | 00 _H | 8 | FFDD0030 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Hall sensor input select register | PIC1ATSGHALLSEL | 00 _H | 8 | FFDD0074 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | TAUD0 input select register | PIC1ATAUD0SEL | 00000000 _H | 32 | FFDD0078 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | TAUD1 input select register | PIC1ATAUD1SEL | 00000000 _H | 32 | FFDD007C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Hi-Z control register 0 | PIC1AHIZCEN0 | 00 _H | 8 | FFDD0080 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Hi-Z control register 1 | PIC1AHIZCEN1 | 00 _H | 8 | FFDD0084 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Hi-Z control register 2 | PIC1AHIZCEN2 | 00 _H | 8 | FFDD0088 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Hi-Z control register 3 | PIC1AHIZCEN3 | 00 _H | 8 | FFDD008C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | ENCATIN1 input select register 400 | PIC1AENCSEL400 | 00 _H | 8 | FFDD00B8 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | ENCATIN1 input select register 410 | PIC1AENCSEL410 | 00 _H | 8 | FFDD00BC _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 200 | PIC1AREG200 | 00000000 _H | 32 | FFDD00C0 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 201 | PIC1AREG201 | 00000000 _H | 32 | FFDD00C4 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 202 | PIC1AREG202 | 00000000 _H | 32 | FFDD00C8 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 203 | PIC1AREG203 | 00000000 _H | 32 | FFDD00CC _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 210 | PIC1AREG210 | 00000000 _H | 32 | FFDD00D4 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 211 | PIC1AREG211 | 00000000 _H | 32 | FFDD00D8 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (82/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PIC1A | Timer input/output control register 212 | PIC1AREG212 | 00000000 _H | 32 | FFDD00DC _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 213 | PIC1AREG213 | 00000000 _H | 32 | FFDD00E0 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 30 | PIC1AREG30 | 00000000 _H | 32 | FFDD00E8 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 31 | PIC1AREG31 | 00000000 _H | 32 | FFDD00EC _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 50 | PIC1AREG50 | 0000 _H | 16 | FFDD00F8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Timer input/output control register 51 | PIC1AREG51 | 0000 _H | 16 | FFDD00FC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 0 trigger select control register 0 | PIC2BADCG0TSEL0 | 00000000 _H | 32 | FFDD1000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 0 trigger select control register 1 | PIC2BADCG0TSEL1 | 00000000 _H | 32 | FFDD1004 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 0 trigger select control register 2 | PIC2BADCG0TSEL2 | 00000000 _H | 32 | FFDD1008 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 0 trigger select control register 3 | PIC2BADCG0TSEL3 | 00000000 _H | 32 | FFDD100C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 0 trigger select control register 4 | PIC2BADCG0TSEL4 | 00000000 _H | 32 | FFDD1010 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 0 trigger edge select control register | PIC2BADCG0EDGSEL | 0000 _H | 16 | FFDD101C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 1 trigger select control register 0 | PIC2BADCG1TSEL0 | 00000000 _H | 32 | FFDD1020 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 1 trigger select control register 1 | PIC2BADCG1TSEL1 | 00000000 _H | 32 | FFDD1024 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 1 trigger select control register 2 | PIC2BADCG1TSEL2 | 00000000 _H | 32 | FFDD1028 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 1 trigger select control register 3 | PIC2BADCG1TSEL3 | 00000000 _H | 32 | FFDD102C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 1 trigger select control register 4 | PIC2BADCG1TSEL4 | 00000000 _H | 32 | FFDD1030 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter 1 trigger edge select control register | PIC2BADCG1EDGSEL | 0000 _H | 16 | FFDD103C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 400 | PIC2BADTEN400 | 0000 _H | 16 | FFDD1040 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 401 | PIC2BADTEN401 | 0000 _H | 16 | FFDD1044 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 402 | PIC2BADTEN402 | 0000 _H | 16 | FFDD1048 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 403 | PIC2BADTEN403 | 0000 _H | 16 | FFDD104C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 404 | PIC2BADTEN404 | 0000 _H | 16 | FFDD1050 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 410 | PIC2BADTEN410 | 0000 _H | 16 | FFDD1060 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 411 | PIC2BADTEN411 | 0000 _H | 16 | FFDD1064 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 412 | PIC2BADTEN412 | 0000 _H | 16 | FFDD1068 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 413 | PIC2BADTEN413 | 0000 _H | 16 | FFDD106C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter trigger output control register 414 | PIC2BADTEN414 | 0000 _H | 16 | FFDD1070 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Synchronous clear enable register | SELBSSER | 00 _H | 8 | FFDD2000 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM | OSTM0 clock select register | ICOCKSEL0 | 0000 _H | 16 | FFDD6000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM | OSTM1 clock select register | ICOCKSEL1 | 0000 _H | 16 | FFDD6004 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Port input monitor select register | PIMONSEL | 00 _H | 8 | FFDD7000 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC1A | Port output monitor select register | POMONSEL | 00 _H | 8 | FFDD7400 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 compare register | OSTM0CMP | 00000000 _H | 32 | FFDD8000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 compare register | OSTM0CNT | FFFFFFF _H | 32 | FFDD8004 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 output register | OSTM0TO | 00 _H | 8 | FFDD8008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 output enable register | OSTM0TOE | 00 _H | 8 | FFDD800C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 count enable status register | OSTM0TE | 00 _H | 8 | FFDD8010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 count start trigger register | OSTM0TS | 00 _H | 8 | FFDD8014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 count stop trigger register | OSTM0TT | 00 _H | 8 | FFDD8018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM0 | OSTM0 control register | OSTM0CTL | 00 _H | 8 | FFDD8020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM1 | OSTM1 compare register | OSTM1CMP | 00000000 _H | 32 | FFDD9000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM1 | OSTM1 counter register | OSTM1CNT | FFFFFFF _H | 32 | FFDD9004 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM1 | OSTM1 output register | OSTM1TO | 00 _H | 8 | FFDD9008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM1 | OSTM1 output enable register | OSTM1TOE | 00 _H | 8 | FFDD900C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM1 | OSTM1 count enable status register | OSTM1TE | 00 _H | 8 | FFDD9010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM1 | OSTM1 count start trigger register | OSTM1TS | 00 _H | 8 | FFDD9014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OSTM1 | OSTM1 count stop trigger register | OSTM1TT | 00 _H | 8 | FFDD9018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (83/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| OSTM1 | OSTM1 control register | OSTM1CTL | 00 _H | 8 | FFDD9020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT00 | 0605FE1B _H | 32 | FFDD0000 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID00 | FFFFFFF _H | 32 | FFDD0004 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT01 | 0605FE17 _H | 32 | FFDD0008 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID01 | FFFFFFF _H | 32 | FFDD000C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT02 | 0605FE1B _H | 32 | FFDD0010 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID02 | FFFFFFF _H | 32 | FFDD0014 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT03 | 0605FE17 _H | 32 | FFDD0018 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID03 | FFFFFFF _H | 32 | FFDD001C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT04 | 0605FE1B _H | 32 | FFDD0020 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID04 | FFFFFFF _H | 32 | FFDD0024 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT05 | 0605FE17 _H | 32 | FFDD0028 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID05 | FFFFFFF _H | 32 | FFDD002C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT06 | 0605FE1B _H | 32 | FFDD0030 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID06 | FFFFFFF _H | 32 | FFDD0034 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT07 | 0605FE17 _H | 32 | FFDD0038 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID07 | FFFFFFF _H | 32 | FFDD003C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT08 | 0605FE1B _H | 32 | FFDD0040 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID08 | FFFFFFF _H | 32 | FFDD0044 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT09 | 0605FE17 _H | 32 | FFDD0048 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID09 | FFFFFFF _H | 32 | FFDD004C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT10 | 0605FE1B _H | 32 | FFDD0050 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID10 | FFFFFFF _H | 32 | FFDD0054 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT11 | 0605FE17 _H | 32 | FFDD0058 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID11 | FFFFFFF _H | 32 | FFDD005C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT12 | 0605FE1B _H | 32 | FFDD0060 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID12 | FFFFFFF _H | 32 | FFDD0064 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT13 | 0605FE17 _H | 32 | FFDD0068 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID13 | FFFFFFF _H | 32 | FFDD006C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT14 | 0605FE1B _H | 32 | FFDD0070 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID14 | FFFFFFF _H | 32 | FFDD0074 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1APROT15 | 0605FE17 _H | 32 | FFDD0078 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ASPID15 | FFFFFFF _H | 32 | FFDD007C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT00 | 0605FE1B _H | 32 | FFDD0080 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID00 | FFFFFFF _H | 32 | FFDD0084 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT01 | 0605FE17 _H | 32 | FFDD0088 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID01 | FFFFFFF _H | 32 | FFDD008C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT02 | 0605FE1B _H | 32 | FFDD0090 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID02 | FFFFFFF _H | 32 | FFDD0094 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT03 | 0605FE17 _H | 32 | FFDD0098 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID03 | FFFFFFF _H | 32 | FFDD009C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT04 | 0605FE1B _H | 32 | FFDD00A0 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID04 | FFFFFFF _H | 32 | FFDD00A4 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT05 | 0605FE17 _H | 32 | FFDD00A8 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID05 | FFFFFFF _H | 32 | FFDD00AC _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT06 | 0605FE1B _H | 32 | FFDD00B0 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID06 | FFFFFFF _H | 32 | FFDD00B4 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT07 | 0605FE17 _H | 32 | FFDD00B8 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (84/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------|-------------------|----------------|---------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID07 | FFFFFFFH | 32 | FFDD0BC | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT08 | 0605FE1B | H 32 | FFDD0C0 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID08 | FFFFFFFH | 32 | FFDD0C4 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT09 | 0605FE17 | H 32 | FFDD0C8 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID09 | FFFFFFFH | 32 | FFDD0CC | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT10 | 0605FE1B | H 32 | FFDD0D0 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID10 | FFFFFFFH | 32 | FFDD0D4 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT11 | 0605FE17 | H 32 | FFDD0D8 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID11 | FFFFFFFH | 32 | FFDD0DC | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT12 | 0605FE1B | H 32 | FFDD0E0 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID12 | FFFFFFFH | 32 | FFDD0E4 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT13 | 0605FE17 | H 32 | FFDD0E8 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID13 | FFFFFFFH | 32 | FFDD0EC | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT14 | 0605FE1B | H 32 | FFDD0F0 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID14 | FFFFFFFH | 32 | FFDD0F4 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1BPROT15 | 0605FE17 | H 32 | FFDD0F8 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1BSPID15 | FFFFFFFH | 32 | FFDD0FC | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT00 | 0605FE1B | H 32 | FFDD100 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID00 | FFFFFFFH | 32 | FFDD104 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT01 | 0605FE17 | H 32 | FFDD108 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID01 | FFFFFFFH | 32 | FFDD10C | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT02 | 0605FE1B | H 32 | FFDD110 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID02 | FFFFFFFH | 32 | FFDD114 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT03 | 0605FE17 | H 32 | FFDD118 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID03 | FFFFFFFH | 32 | FFDD11C | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT04 | 0605FE1B | H 32 | FFDD120 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID04 | FFFFFFFH | 32 | FFDD124 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT05 | 0605FE17 | H 32 | FFDD128 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID05 | FFFFFFFH | 32 | FFDD12C | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT06 | 0605FE1B | H 32 | FFDD130 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID06 | FFFFFFFH | 32 | FFDD134 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT07 | 0605FE17 | H 32 | FFDD138 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID07 | FFFFFFFH | 32 | FFDD13C | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT08 | 0605FE1B | H 32 | FFDD140 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID08 | FFFFFFFH | 32 | FFDD144 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT09 | 0605FE17 | H 32 | FFDD148 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID09 | FFFFFFFH | 32 | FFDD14C | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT10 | 0605FE1B | H 32 | FFDD150 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID10 | FFFFFFFH | 32 | FFDD154 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT11 | 0605FE17 | H 32 | FFDD158 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID11 | FFFFFFFH | 32 | FFDD15C | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT12 | 0605FE1B | H 32 | FFDD160 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID12 | FFFFFFFH | 32 | FFDD164 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT13 | 0605FE17 | H 32 | FFDD168 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID13 | FFFFFFFH | 32 | FFDD16C | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT14 | 0605FE1B | H 32 | FFDD170 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID14 | FFFFFFFH | 32 | FFDD174 | H 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1CPROT15 | 0605FE17 | H 32 | FFDD178 | H 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (85/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1CSPID15 | FFFFFFFF _H | 32 | FFDDD17C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT00 | 0605FE1B _H | 32 | FFDDD180 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID00 | FFFFFFFF _H | 32 | FFDDD184 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT01 | 0605FE17 _H | 32 | FFDDD188 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID01 | FFFFFFFF _H | 32 | FFDDD18C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT02 | 0605FE1B _H | 32 | FFDDD190 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID02 | FFFFFFFF _H | 32 | FFDDD194 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT03 | 0605FE17 _H | 32 | FFDDD198 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID03 | FFFFFFFF _H | 32 | FFDDD19C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT04 | 0605FE1B _H | 32 | FFDDD1A0 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID04 | FFFFFFFF _H | 32 | FFDDDA14 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT05 | 0605FE17 _H | 32 | FFDDDA18 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID05 | FFFFFFFF _H | 32 | FFDDDA1C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT06 | 0605FE1B _H | 32 | FFDDDA10 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID06 | FFFFFFFF _H | 32 | FFDDDA14 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT07 | 0605FE17 _H | 32 | FFDDDA18 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID07 | FFFFFFFF _H | 32 | FFDDDA1C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT08 | 0605FE1B _H | 32 | FFDDDA10 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID08 | FFFFFFFF _H | 32 | FFDDDA14 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT09 | 0605FE17 _H | 32 | FFDDDA18 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID09 | FFFFFFFF _H | 32 | FFDDDA1C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT10 | 0605FE1B _H | 32 | FFDDDA10 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID10 | FFFFFFFF _H | 32 | FFDDDA14 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT11 | 0605FE17 _H | 32 | FFDDDA18 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID11 | FFFFFFFF _H | 32 | FFDDDA1C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT12 | 0605FE1B _H | 32 | FFDDDA10 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID12 | FFFFFFFF _H | 32 | FFDDDA14 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT13 | 0605FE17 _H | 32 | FFDDDA18 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID13 | FFFFFFFF _H | 32 | FFDDDA1C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT14 | 0605FE1B _H | 32 | FFDDDA10 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID14 | FFFFFFFF _H | 32 | FFDDDA14 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1DPROT15 | 0605FE17 _H | 32 | FFDDDA18 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1DSPID15 | FFFFFFFF _H | 32 | FFDDDA1C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT00 | 0605FE1B _H | 32 | FFDDD200 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID00 | FFFFFFFF _H | 32 | FFDDD204 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT01 | 0605FE17 _H | 32 | FFDDD208 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID01 | FFFFFFFF _H | 32 | FFDDD20C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT02 | 0605FE1B _H | 32 | FFDDD210 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID02 | FFFFFFFF _H | 32 | FFDDD214 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT03 | 0605FE17 _H | 32 | FFDDD218 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID03 | FFFFFFFF _H | 32 | FFDDD21C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT04 | 0605FE1B _H | 32 | FFDDD220 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID04 | FFFFFFFF _H | 32 | FFDDD224 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT05 | 0605FE17 _H | 32 | FFDDD228 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID05 | FFFFFFFF _H | 32 | FFDDD22C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT06 | 0605FE1B _H | 32 | FFDDD230 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID06 | FFFFFFFF _H | 32 | FFDDD234 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT07 | 0605FE17 _H | 32 | FFDDD238 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (86/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------|-----------------------|----------------|----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID07 | FFFFFFFH _H | 32 | FFDD23C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT08 | 0605FE1B _H | 32 | FFDD240 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID08 | FFFFFFFH _H | 32 | FFDD244 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT09 | 0605FE17 _H | 32 | FFDD248 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID09 | FFFFFFFH _H | 32 | FFDD24C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT10 | 0605FE1B _H | 32 | FFDD250 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID10 | FFFFFFFH _H | 32 | FFDD254 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT11 | 0605FE17 _H | 32 | FFDD258 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID11 | FFFFFFFH _H | 32 | FFDD25C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT12 | 0605FE1B _H | 32 | FFDD260 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID12 | FFFFFFFH _H | 32 | FFDD264 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT13 | 0605FE17 _H | 32 | FFDD268 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID13 | FFFFFFFH _H | 32 | FFDD26C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT14 | 0605FE1B _H | 32 | FFDD270 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID14 | FFFFFFFH _H | 32 | FFDD274 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-Bus Guard Protection Setting Register | FSGD1EPROT15 | 0605FE17 _H | 32 | FFDD278 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | P-bus Guard SPID Setting Register | FSGD1ESPID15 | FFFFFFFH _H | 32 | FFDD27C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | ERRSLV Control Register for Pbus Guard | ERRSLV1CTL | 00 _H | 8 | FFDD400 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | ERRSLV Status Register for PBus Guard | ERRSLV1STAT | 00000000 _H | 32 | FFDD404 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG1 | ERRSLV Error Transfer Type Register for P-Bus Guard | ERRSLV1TYPE | 00000000 _H | 32 | FFDD40C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC1 | P-Bus Data and Address ECC Control Register | APEC0ECCCTL | 00000000 _H | 32 | FFDD800 _H | 1 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC1 | P-Bus Error Information Control Register | APEC0ERRINT | 00000073 _H | 32 | FFDD804 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC1 | P-Bus Data and Address ECC SED/DED Status Clear Register | APEC0STCLR | 00000000 _H | 32 | FFDD808 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC1 | P-Bus Data and Address Error Count Overflow Status Register | APEC0OVFSTR | 00000000 _H | 32 | FFDD80C _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC1 | P-Bus Data and Address ECC SED/DED Status Register | APEC01STERSTR | 00000000 _H | 32 | FFDD810 _H | 1 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC1 | P-Bus Data and Address ECC SED/DED Address Register | APEC01STEADR0 | 00000000 _H | 32 | FFDD850 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Serial mode register | SCI30SMR | 00 _H | 8 | FFDF000 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Bit rate register | SCI30BRR | FF _H | 8 | FFDF004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Modulation duty register | SCI30MDDR | FF _H | 8 | FFDF004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Serial control register | SCI30SCR | 00 _H | 8 | FFDF008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Transmit data register | SCI30TDR | FF _H | 8 | FFDF00C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Serial status register | SCI30SSR | 84 _H | 8 | FFDF010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Receive data register | SCI30RDR | 00 _H | 8 | FFDF014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Serial transfer format register | SCI30SCMR | F2 _H | 8 | FFDF018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI30 | Serial extended mode register | SCI30SEMR | 04 _H | 8 | FFDF01C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Serial mode register | SCI31SMR | 00 _H | 8 | FFDF100 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Bit rate register | SCI31BRR | FF _H | 8 | FFDF104 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Modulation duty register | SCI31MDDR | FF _H | 8 | FFDF104 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Serial control register | SCI31SCR | 00 _H | 8 | FFDF108 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Transmit data register | SCI31TDR | FF _H | 8 | FFDF10C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Serial status register | SCI31SSR | 84 _H | 8 | FFDF110 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Receive data register | SCI31RDR | 00 _H | 8 | FFDF114 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Serial transfer format register | SCI31SCMR | F2 _H | 8 | FFDF118 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI31 | Serial extended mode register | SCI31SEMR | 04 _H | 8 | FFDF11C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Serial mode register | SCI32SMR | 00 _H | 8 | FFDF200 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Bit rate register | SCI32BRR | FF _H | 8 | FFDF204 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (87/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| SCI32 | Modulation duty register | SCI32MDDR | FF _H | 8 | FFDF2004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Serial control register | SCI32SCR | 00 _H | 8 | FFDF2008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Transmit data register | SCI32TDR | FF _H | 8 | FFDF200C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Serial status register | SCI32SSR | 84 _H | 8 | FFDF2010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Receive data register | SCI32RDR | 00 _H | 8 | FFDF2014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Serial transfer format register | SCI32SCMR | F2 _H | 8 | FFDF2018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SCI32 | Serial extended mode register | SCI32SEMR | 04 _H | 8 | FFDF201C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN wake-up baud rate selector register | RLN30LWBR | 00 _H | 8 | FFDF8001 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART baud rate prescaler 01 register | RLN30LBRP01 | 0000 _H | 16 | FFDF8002 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART baud rate prescaler 0 register | RLN30LBRP0 | 00 _H | 8 | FFDF8002 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART baud rate prescaler 1 register | RLN30LBRP1 | 00 _H | 8 | FFDF8003 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN self-test control register | RLN30LSTC | 00 _H | 8 | FFDF8004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART mode register | RLN30LMD | 00 _H | 8 | FFDF8008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN break field configuration register/ UART configuration register | RLN30LBFC | 00 _H | 8 | FFDF8009 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART space configuration register | RLN30LSC | 00 _H | 8 | FFDF800A _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN wake-up configuration register | RLN30LWUP | 00 _H | 8 | FFDF800B _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN interrupt enable register | RLN30LIE | 00 _H | 8 | FFDF800C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART error detection enable register | RLN30LEDE | 00 _H | 8 | FFDF800D _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART control register | RLN30LCUC | 00 _H | 8 | FFDF800E _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART transmission control register | RLN30LTRC | 00 _H | 8 | FFDF8010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART mode status register | RLN30LMST | 00 _H | 8 | FFDF8011 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART status register | RLN30LST | 00 _H | 8 | FFDF8012 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART error status register | RLN30LEST | 00 _H | 8 | FFDF8013 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN data field configuration register | RLN30LDFC | 00 _H | 8 | FFDF8014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART ID buffer register | RLN30LIDB | 00 _H | 8 | FFDF8015 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN checksum buffer register | RLN30LCBR | 00 _H | 8 | FFDF8016 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | UART data 0 buffer register | RLN30LUDB0 | 00 _H | 8 | FFDF8017 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART data buffer 1 register | RLN30LDBR1 | 00 _H | 8 | FFDF8018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART data buffer 2 register | RLN30LDBR2 | 00 _H | 8 | FFDF8019 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART data buffer 3 register | RLN30LDBR3 | 00 _H | 8 | FFDF801A _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART data buffer 4 register | RLN30LDBR4 | 00 _H | 8 | FFDF801B _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART data buffer 5 register | RLN30LDBR5 | 00 _H | 8 | FFDF801C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART data buffer 6 register | RLN30LDBR6 | 00 _H | 8 | FFDF801D _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN / UART data buffer 7 register | RLN30LDBR7 | 00 _H | 8 | FFDF801E _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | LIN/ UART data buffer 8 register | RLN30LDBR8 | 00 _H | 8 | FFDF801F _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | UART operation enable register | RLN30LUOER | 00 _H | 8 | FFDF8020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | UART option register 1 | RLN30LUOR1 | 00 _H | 8 | FFDF8021 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | UART transmission data register | RLN30LUTDR | 0000 _H | 16 | FFDF8024 _H | 1 | 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | UART reception data register | RLN30LURDR | 0000 _H | 16 | FFDF8026 _H | 1 | 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN30 | UART wait transmission data register | RLN30LUWTD | 0000 _H | 16 | FFDF8028 _H | 1 | 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN wake-up baud rate selector register | RLN31LWBR | 00 _H | 8 | FFDF9001 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART baud rate prescaler 01 register | RLN31LBRP01 | 0000 _H | 16 | FFDF9002 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART baud rate prescaler 0 register | RLN31LBRP0 | 00 _H | 8 | FFDF9002 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART baud rate prescaler 1 register | RLN31LBRP1 | 00 _H | 8 | FFDF9003 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN self-test control register | RLN31LSTC | 00 _H | 8 | FFDF9004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART mode register | RLN31LMD | 00 _H | 8 | FFDF9008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN break field configuration register/ UART configuration register | RLN31LBFC | 00 _H | 8 | FFDF9009 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (88/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RLN31 | LIN / UART space configuration register | RLN31LSC | 00 _H | 8 | FFDF900A _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN wake-up configuration register | RLN31LWUP | 00 _H | 8 | FFDF900B _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN interrupt enable register | RLN31LIE | 00 _H | 8 | FFDF900C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART error detection enable register | RLN31LEDE | 00 _H | 8 | FFDF900D _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART control register | RLN31LCUC | 00 _H | 8 | FFDF900E _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART transmission control register | RLN31LTRC | 00 _H | 8 | FFDF9010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART mode status register | RLN31LMST | 00 _H | 8 | FFDF9011 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART status register | RLN31LST | 00 _H | 8 | FFDF9012 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART error status register | RLN31LEST | 00 _H | 8 | FFDF9013 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN data field configuration register | RLN31LDFC | 00 _H | 8 | FFDF9014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART ID buffer register | RLN31LIDB | 00 _H | 8 | FFDF9015 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN checksum buffer register | RLN31LCBR | 00 _H | 8 | FFDF9016 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | UART data 0 buffer register | RLN31LUDB0 | 00 _H | 8 | FFDF9017 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART data buffer 1 register | RLN31LDBR1 | 00 _H | 8 | FFDF9018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART data buffer 2 register | RLN31LDBR2 | 00 _H | 8 | FFDF9019 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART data buffer 3 register | RLN31LDBR3 | 00 _H | 8 | FFDF901A _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART data buffer 4 register | RLN31LDBR4 | 00 _H | 8 | FFDF901B _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART data buffer 5 register | RLN31LDBR5 | 00 _H | 8 | FFDF901C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART data buffer 6 register | RLN31LDBR6 | 00 _H | 8 | FFDF901D _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN / UART data buffer 7 register | RLN31LDBR7 | 00 _H | 8 | FFDF901E _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | LIN/ UART data buffer 8 register | RLN31LDBR8 | 00 _H | 8 | FFDF901F _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | UART operation enable register | RLN31LUOER | 00 _H | 8 | FFDF9020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | UART option register 1 | RLN31LUOR1 | 00 _H | 8 | FFDF9021 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | UART transmission data register | RLN31LUTDR | 0000 _H | 16 | FFDF9024 _H | 1 | 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | UART reception data register | RLN31LURDR | 0000 _H | 16 | FFDF9026 _H | 1 | 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLN31 | UART wait transmission data register | RLN31LUWTD | 0000 _H | 16 | FFDF9028 _H | 1 | 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 channel control register | PSI50CHCTRL | 00000000 _H | 32 | FFE00000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 IP timer control register | PSI50PTIMERCTRL | 00010000 _H | 32 | FFE00010 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 IP timer counter | PSI50PTIMER | 00000000 _H | 32 | FFE00014 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 operating mode/communication mode register | PSI50OPMCOMM | 00000000 _H | 32 | FFE00020 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 operating-mode bit rate register | PSI50OPMBITRATE | 0000029F _H | 32 | FFE00024 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 operating-mode cycle time register | PSI50OPMCYCT | 00009C3F _H | 32 | FFE00028 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 interrupt status register | PSI50PSI5INT | 00000000 _H | 32 | FFE00030 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive data emulation register | PSI50EMRXDATA | 00000000 _H | 32 | FFE00040 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive data status emulation register | PSI50EMRXDST | 00000000 _H | 32 | FFE00044 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive data IP timer emulation register | PSI50EMRXDTIM | 00000000 _H | 32 | FFE00048 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive data FIFO emulation register | PSI50EMRXDFIFO | 00000000 _H | 32 | FFE0004C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive-message receive message emulation register | PSI50EMRXMRMSG | 00000000 _H | 32 | FFE00050 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive-message channel receive status emulation register | PSI50EMRXMRXST | 00000000 _H | 32 | FFE00054 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive-message channel receive timestamp emulation register | PSI50EMRXMRXTIM | 00000000 _H | 32 | FFE00058 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 receive-message channel FIFO emulation register | PSI50EMRXMFIFO | 00000000 _H | 32 | FFE0005C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 transmission setting register | PSI50TXSETTING | 106747F7 _H | 32 | FFE00080 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 synchronization control register | PSI50SYNCCTRL | 00000000 _H | 32 | FFE00084 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 transmission status register | PSI50TXST | 00000001 _H | 32 | FFE00088 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 transmission status clear register | PSI50TXSTCLR | 00000000 _H | 32 | FFE0008C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI50 | PSI5 transmission status interrupt enable register | PSI50TXSTINTEN | 00000000 _H | 32 | FFE00090 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (89/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PSI50 | PSI5 transmit data control register | PSI50TXDCTRL | 00000001 _H | 32 | FFE00094 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 transmit data register | PSI50TXDATA | 00000000 _H | 32 | FFE00098 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive sampling setting register | PSI50RXSPLSET | 00000027 _H | 32 | FFE00100 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 1 setting register | PSI50RXS1SET | 00A00000 _H | 32 | FFE00108 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 2 setting register | PSI50RXS2SET | 00A00000 _H | 32 | FFE0010C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 3 setting register | PSI50RXS3SET | 00A00000 _H | 32 | FFE00110 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 4 setting register | PSI50RXS4SET | 00A00000 _H | 32 | FFE00114 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 5 setting register | PSI50RXS5SET | 00A00000 _H | 32 | FFE00118 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 6 setting register | PSI50RXS6SET | 00A00000 _H | 32 | FFE0011C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 7 setting register | PSI50RXS7SET | 00A00000 _H | 32 | FFE00120 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive slot 8 setting register | PSI50RXS8SET | 00A00000 _H | 32 | FFE00124 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive data register | PSI50RXDATA | 00000000 _H | 32 | FFE00128 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive data status register | PSI50RXDST | 00000000 _H | 32 | FFE0012C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive data IP timer register | PSI50RXDTIM | 00000000 _H | 32 | FFE00130 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive data FIFO register | PSI50RXDFIFO | 00000000 _H | 32 | FFE00134 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive module status register | PSI50RXMODST | 00000000 _H | 32 | FFE00138 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive module status clear register | PSI50RXMODSTCLR | 00000000 _H | 32 | FFE0013C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive module status interrupt enable register | PSI50RXMODSTINTEN | 00000000 _H | 32 | FFE00140 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive message channel setting register | PSI50RXMSET | 00000000 _H | 32 | FFE00180 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive-message receive message register | PSI50RXMRMSG | 00000000 _H | 32 | FFE00184 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive-message channel receive status register | PSI50RXMRXST | 00000000 _H | 32 | FFE00188 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive-message channel receive timestamp register | PSI50RXMRXTIM | 00000000 _H | 32 | FFE0018C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive-message channel FIFO register | PSI50RXMFIFO | 00000000 _H | 32 | FFE00190 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive-message channel module status register | PSI50RXMMST | 00000000 _H | 32 | FFE00194 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive-message channel module status clear register | PSI50RXMMSTCLR | 00000000 _H | 32 | FFE00198 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI50 | PSI5 receive-message channel module status interrupt enable register | PSI50RXMMSTINTEN | 00000000 _H | 32 | FFE0019C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 channel control register | PSI51CHCTRL | 00000000 _H | 32 | FFE01000 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 IP timer control register | PSI51IPTIMERCTRL | 00010000 _H | 32 | FFE01010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 IP timer counter | PSI51IPTIMER | 00000000 _H | 32 | FFE01014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 operating mode/communication mode register | PSI51OPMCOMM | 00000000 _H | 32 | FFE01020 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 operating-mode bit rate register | PSI51OPMBITRATE | 0000029F _H | 32 | FFE01024 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 operating-mode cycle time register | PSI51OPMCYCT | 00009C3F _H | 32 | FFE01028 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 interrupt status register | PSI51PSI5INT | 00000000 _H | 32 | FFE01030 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive data emulation register | PSI51EMRXDATA | 00000000 _H | 32 | FFE01040 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive data status emulation register | PSI51EMRXDST | 00000000 _H | 32 | FFE01044 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive data IP timer emulation register | PSI51EMRXDTIM | 00000000 _H | 32 | FFE01048 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive data FIFO emulation register | PSI51EMRXDFIFO | 00000000 _H | 32 | FFE0104C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive-message receive message emulation register | PSI51EMRXMRMSG | 00000000 _H | 32 | FFE01050 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive-message channel receive status emulation register | PSI51EMRXMRXST | 00000000 _H | 32 | FFE01054 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive-message channel receive timestamp emulation register | PSI51EMRXMRXTIM | 00000000 _H | 32 | FFE01058 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 receive-message channel FIFO emulation register | PSI51EMRXMFIFO | 00000000 _H | 32 | FFE0105C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 transmission setting register | PSI51TXSETTING | 106747F7 _H | 32 | FFE01080 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 synchronization control register | PSI51SYNCCTRL | 00000000 _H | 32 | FFE01084 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| PSI51 | PSI5 transmission status register | PSI51TXST | 00000001 _H | 32 | FFE01088 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (90/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PSI51 | PSI5 transmission status clear register | PSI51TXSTCLR | 00000000 _H | 32 | FFE0108C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 transmission status interrupt enable register | PSI51TXSTINTEN | 00000000 _H | 32 | FFE01090 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 transmit data control register | PSI51TXDCTRL | 00000001 _H | 32 | FFE01094 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 transmit data register | PSI51TXDATA | 00000000 _H | 32 | FFE01098 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive sampling setting register | PSI51RXSPLSET | 00000027 _H | 32 | FFE01100 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 1 setting register | PSI51RXS1SET | 00A00000 _H | 32 | FFE01108 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 2 setting register | PSI51RXS2SET | 00A00000 _H | 32 | FFE0110C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 3 setting register | PSI51RXS3SET | 00A00000 _H | 32 | FFE01110 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 4 setting register | PSI51RXS4SET | 00A00000 _H | 32 | FFE01114 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 5 setting register | PSI51RXS5SET | 00A00000 _H | 32 | FFE01118 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 6 setting register | PSI51RXS6SET | 00A00000 _H | 32 | FFE0111C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 7 setting register | PSI51RXS7SET | 00A00000 _H | 32 | FFE01120 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive slot 8 setting register | PSI51RXS8SET | 00A00000 _H | 32 | FFE01124 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive data register | PSI51RXDATA | 00000000 _H | 32 | FFE01128 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive data status register | PSI51RXDST | 00000000 _H | 32 | FFE0112C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive data IP timer register | PSI51RXDTIM | 00000000 _H | 32 | FFE01130 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive data FIFO register | PSI51RXDFIFO | 00000000 _H | 32 | FFE01134 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive module status register | PSI51RXMODST | 00000000 _H | 32 | FFE01138 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive module status clear register | PSI51RXMODSTCLR | 00000000 _H | 32 | FFE0113C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive module status interrupt enable register | PSI51RXMODSTINTEN | 00000000 _H | 32 | FFE01140 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive message channel setting register | PSI51RXMSET | 00000000 _H | 32 | FFE01180 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive-message receive message register | PSI51RXMRXMSG | 00000000 _H | 32 | FFE01184 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive-message channel receive status register | PSI51RXMRXST | 00000000 _H | 32 | FFE01188 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive-message channel receive timestamp register | PSI51RXMRXTIM | 00000000 _H | 32 | FFE0118C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive-message channel FIFO register | PSI51RXMFIFO | 00000000 _H | 32 | FFE01190 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive-message channel module status register | PSI51RXMMST | 00000000 _H | 32 | FFE01194 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive-message channel module status clear register | PSI51RXMMSTCLR | 00000000 _H | 32 | FFE01198 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI51 | PSI5 receive-message channel module status interrupt enable register | PSI51RXMMSTINTEN | 00000000 _H | 32 | FFE0119C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PSI5 | PSI5 timestamp function mode selection register | PSI5TSSEL | 00000000 _H | 32 | FFE03000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT timestamp register | RSENT0TSPC | 00000000 _H | 32 | FFE05000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT timestamp counter | RSENT0TSC | 00000000 _H | 32 | FFE05004 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT communication configuration register | RSENT0CC | 00000000 _H | 32 | FFE05010 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT baud rate prescaler register | RSENT0BRP | 00000000 _H | 32 | FFE05014 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT interrupt/DMA enable register | RSENT0IDE | 00000000 _H | 32 | FFE05018 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT mode control register | RSENT0MDC | 00000000 _H | 32 | FFE0501C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT SPC transmission register | RSENT0SPCT | 00000000 _H | 32 | FFE05020 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT mode status register | RSENT0MST | 00000000 _H | 32 | FFE05024 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT communication status register | RSENT0CS | 00000000 _H | 32 | FFE05028 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT communication status clear register | RSENT0CSC | 00000000 _H | 32 | FFE0502C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT slow channel receive timestamp register | RSENT0SRTS | 00000000 _H | 32 | FFE05030 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT slow channel receive data register | RSENT0SRXD | 00000000 _H | 32 | FFE05034 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT calibration pulse length register | RSENT0CPL | 00000000 _H | 32 | FFE05038 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT message length register | RSENT0ML | 00000000 _H | 32 | FFE0503C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT fast channel receive timestamp register | RSENT0FRTS | 00000000 _H | 32 | FFE05040 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT0 | RSENT fast channel receive data register | RSENT0FRXD | 00000000 _H | 32 | FFE05044 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| RSENT1 | RSENT timestamp register | RSENT1TSPC | 00000000 _H | 32 | FFE06000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (91/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSENT1 | RSENT timestamp counter | RSENT1TSC | 00000000 _H | 32 | FFE06004 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT communication configuration register | RSENT1CC | 00000000 _H | 32 | FFE06010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT baud rate prescaler register | RSENT1BRP | 00000000 _H | 32 | FFE06014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT interrupt/DMA enable register | RSENT1IDE | 00000000 _H | 32 | FFE06018 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT mode control register | RSENT1MDC | 00000000 _H | 32 | FFE0601C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT SPC transmission register | RSENT1SPCT | 00000000 _H | 32 | FFE06020 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT mode status register | RSENT1MST | 00000000 _H | 32 | FFE06024 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT communication status register | RSENT1CS | 00000000 _H | 32 | FFE06028 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT communication status clear register | RSENT1CSC | 00000000 _H | 32 | FFE0602C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT slow channel receive timestamp register | RSENT1SRTS | 00000000 _H | 32 | FFE06030 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT slow channel receive data register | RSENT1SRXD | 00000000 _H | 32 | FFE06034 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT calibration pulse length register | RSENT1CPL | 00000000 _H | 32 | FFE06038 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT message length register | RSENT1ML | 00000000 _H | 32 | FFE0603C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT fast channel receive timestamp register | RSENT1FRTS | 00000000 _H | 32 | FFE06040 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT1 | RSENT fast channel receive data register | RSENT1FRXD | 00000000 _H | 32 | FFE06044 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT timestamp register | RSENT2TSPC | 00000000 _H | 32 | FFE07000 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT timestamp counter | RSENT2TSC | 00000000 _H | 32 | FFE07004 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT communication configuration register | RSENT2CC | 00000000 _H | 32 | FFE07010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT baud rate prescaler register | RSENT2BRP | 00000000 _H | 32 | FFE07014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT interrupt/DMA enable register | RSENT2IDE | 00000000 _H | 32 | FFE07018 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT mode control register | RSENT2MDC | 00000000 _H | 32 | FFE0701C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT SPC transmission register | RSENT2SPCT | 00000000 _H | 32 | FFE07020 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT mode status register | RSENT2MST | 00000000 _H | 32 | FFE07024 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT communication status register | RSENT2CS | 00000000 _H | 32 | FFE07028 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT communication status clear register | RSENT2CSC | 00000000 _H | 32 | FFE0702C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT slow channel receive timestamp register | RSENT2SRTS | 00000000 _H | 32 | FFE07030 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT slow channel receive data register | RSENT2SRXD | 00000000 _H | 32 | FFE07034 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT calibration pulse length register | RSENT2CPL | 00000000 _H | 32 | FFE07038 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT message length register | RSENT2ML | 00000000 _H | 32 | FFE0703C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT fast channel receive timestamp register | RSENT2FRTS | 00000000 _H | 32 | FFE07040 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT2 | RSENT fast channel receive data register | RSENT2FRXD | 00000000 _H | 32 | FFE07044 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT timestamp register | RSENT3TSPC | 00000000 _H | 32 | FFE08000 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT timestamp counter | RSENT3TSC | 00000000 _H | 32 | FFE08004 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT communication configuration register | RSENT3CC | 00000000 _H | 32 | FFE08010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT baud rate prescaler register | RSENT3BRP | 00000000 _H | 32 | FFE08014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT interrupt/DMA enable register | RSENT3IDE | 00000000 _H | 32 | FFE08018 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT mode control register | RSENT3MDC | 00000000 _H | 32 | FFE0801C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT SPC transmission register | RSENT3SPCT | 00000000 _H | 32 | FFE08020 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT mode status register | RSENT3MST | 00000000 _H | 32 | FFE08024 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT communication status register | RSENT3CS | 00000000 _H | 32 | FFE08028 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT communication status clear register | RSENT3CSC | 00000000 _H | 32 | FFE0802C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT slow channel receive timestamp register | RSENT3SRTS | 00000000 _H | 32 | FFE08030 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT slow channel receive data register | RSENT3SRXD | 00000000 _H | 32 | FFE08034 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT calibration pulse length register | RSENT3CPL | 00000000 _H | 32 | FFE08038 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT message length register | RSENT3ML | 00000000 _H | 32 | FFE0803C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT fast channel receive timestamp register | RSENT3FRTS | 00000000 _H | 32 | FFE08040 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT3 | RSENT fast channel receive data register | RSENT3FRXD | 00000000 _H | 32 | FFE08044 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT timestamp register | RSENT4TSPC | 00000000 _H | 32 | FFE09000 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (92/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| RSENT4 | RSENT timestamp counter | RSENT4TSC | 00000000 _H | 32 | FFE09004 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT communication configuration register | RSENT4CC | 00000000 _H | 32 | FFE09010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT baud rate prescaler register | RSENT4BRP | 00000000 _H | 32 | FFE09014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT interrupt/DMA enable register | RSENT4IDE | 00000000 _H | 32 | FFE09018 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT mode control register | RSENT4MDC | 00000000 _H | 32 | FFE0901C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT SPC transmission register | RSENT4SPCT | 00000000 _H | 32 | FFE09020 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT mode status register | RSENT4MST | 00000000 _H | 32 | FFE09024 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT communication status register | RSENT4CS | 00000000 _H | 32 | FFE09028 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT communication status clear register | RSENT4CSC | 00000000 _H | 32 | FFE0902C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT slow channel receive timestamp register | RSENT4SRTS | 00000000 _H | 32 | FFE09030 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT slow channel receive data register | RSENT4SRXD | 00000000 _H | 32 | FFE09034 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT calibration pulse length register | RSENT4CPL | 00000000 _H | 32 | FFE09038 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT message length register | RSENT4ML | 00000000 _H | 32 | FFE0903C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT fast channel receive timestamp register | RSENT4FRTS | 00000000 _H | 32 | FFE09040 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT4 | RSENT fast channel receive data register | RSENT4FRXD | 00000000 _H | 32 | FFE09044 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT timestamp register | RSENT5TSPC | 00000000 _H | 32 | FFE0A000 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT timestamp counter | RSENT5TSC | 00000000 _H | 32 | FFE0A004 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT communication configuration register | RSENT5CC | 00000000 _H | 32 | FFE0A010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT baud rate prescaler register | RSENT5BRP | 00000000 _H | 32 | FFE0A014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT interrupt/DMA enable register | RSENT5IDE | 00000000 _H | 32 | FFE0A018 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT mode control register | RSENT5MDC | 00000000 _H | 32 | FFE0A01C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT SPC transmission register | RSENT5SPCT | 00000000 _H | 32 | FFE0A020 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT mode status register | RSENT5MST | 00000000 _H | 32 | FFE0A024 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT communication status register | RSENT5CS | 00000000 _H | 32 | FFE0A028 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT communication status clear register | RSENT5CSC | 00000000 _H | 32 | FFE0A02C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT slow channel receive timestamp register | RSENT5SRTS | 00000000 _H | 32 | FFE0A030 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT slow channel receive data register | RSENT5SRXD | 00000000 _H | 32 | FFE0A034 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT calibration pulse length register | RSENT5CPL | 00000000 _H | 32 | FFE0A038 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT message length register | RSENT5ML | 00000000 _H | 32 | FFE0A03C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT fast channel receive timestamp register | RSENT5FRTS | 00000000 _H | 32 | FFE0A040 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT5 | RSENT fast channel receive data register | RSENT5FRXD | 00000000 _H | 32 | FFE0A044 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| RSENT | RSENT timestamp mode selection register | RSENTTSEL | 00000000 _H | 32 | FFE0F000 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 0 | TAUD0CDR0 | 0000 _H | 16 | FFE20000 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 1 | TAUD0CDR1 | 0000 _H | 16 | FFE20004 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 2 | TAUD0CDR2 | 0000 _H | 16 | FFE20008 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 3 | TAUD0CDR3 | 0000 _H | 16 | FFE2000C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 4 | TAUD0CDR4 | 0000 _H | 16 | FFE20010 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 5 | TAUD0CDR5 | 0000 _H | 16 | FFE20014 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 6 | TAUD0CDR6 | 0000 _H | 16 | FFE20018 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 7 | TAUD0CDR7 | 0000 _H | 16 | FFE2001C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 8 | TAUD0CDR8 | 0000 _H | 16 | FFE20020 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 9 | TAUD0CDR9 | 0000 _H | 16 | FFE20024 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 10 | TAUD0CDR10 | 0000 _H | 16 | FFE20028 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 11 | TAUD0CDR11 | 0000 _H | 16 | FFE2002C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 12 | TAUD0CDR12 | 0000 _H | 16 | FFE20030 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 13 | TAUD0CDR13 | 0000 _H | 16 | FFE20034 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 14 | TAUD0CDR14 | 0000 _H | 16 | FFE20038 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel data register 15 | TAUD0CDR15 | 0000 _H | 16 | FFE2003C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (93/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD0 | TAUD0 channel output active level register | TAUD0TOL | 0000 _H | 16 | FFE20040 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel reload data trigger register | TAUD0RDT | 0000 _H | 16 | FFE20044 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel reload status register | TAUD0RSF | 0000 _H | 16 | FFE20048 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel real-time output register | TAUD0TRO | 0000 _H | 16 | FFE2004C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel modulation output enable register | TAUD0TME | 0000 _H | 16 | FFE20050 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel dead time output level register | TAUD0TDL | 0000 _H | 16 | FFE20054 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel output register | TAUD0TO | 0000 _H | 16 | FFE20058 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel output enable register | TAUD0TOE | 0000 _H | 16 | FFE2005C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 0 | TAUD0CNT0 | FFFF _H | 16 | FFE20080 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 1 | TAUD0CNT1 | FFFF _H | 16 | FFE20084 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 2 | TAUD0CNT2 | FFFF _H | 16 | FFE20088 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 3 | TAUD0CNT3 | FFFF _H | 16 | FFE2008C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 4 | TAUD0CNT4 | FFFF _H | 16 | FFE20090 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 5 | TAUD0CNT5 | FFFF _H | 16 | FFE20094 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 6 | TAUD0CNT6 | FFFF _H | 16 | FFE20098 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 7 | TAUD0CNT7 | FFFF _H | 16 | FFE2009C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 8 | TAUD0CNT8 | FFFF _H | 16 | FFE200A0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 9 | TAUD0CNT9 | FFFF _H | 16 | FFE200A4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 10 | TAUD0CNT10 | FFFF _H | 16 | FFE200A8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 11 | TAUD0CNT11 | FFFF _H | 16 | FFE200AC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 12 | TAUD0CNT12 | FFFF _H | 16 | FFE200B0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 13 | TAUD0CNT13 | FFFF _H | 16 | FFE200B4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 14 | TAUD0CNT14 | FFFF _H | 16 | FFE200B8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel counter register 15 | TAUD0CNT15 | FFFF _H | 16 | FFE200BC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 0 | TAUD0CMUR0 | 00 _H | 8 | FFE200C0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 1 | TAUD0CMUR1 | 00 _H | 8 | FFE200C4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 2 | TAUD0CMUR2 | 00 _H | 8 | FFE200C8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 3 | TAUD0CMUR3 | 00 _H | 8 | FFE200CC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 4 | TAUD0CMUR4 | 00 _H | 8 | FFE200D0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 5 | TAUD0CMUR5 | 00 _H | 8 | FFE200D4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 6 | TAUD0CMUR6 | 00 _H | 8 | FFE200D8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 7 | TAUD0CMUR7 | 00 _H | 8 | FFE200DC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 8 | TAUD0CMUR8 | 00 _H | 8 | FFE200E0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 9 | TAUD0CMUR9 | 00 _H | 8 | FFE200E4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 10 | TAUD0CMUR10 | 00 _H | 8 | FFE200E8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 11 | TAUD0CMUR11 | 00 _H | 8 | FFE200EC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 12 | TAUD0CMUR12 | 00 _H | 8 | FFE200F0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 13 | TAUD0CMUR13 | 00 _H | 8 | FFE200F4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 14 | TAUD0CMUR14 | 00 _H | 8 | FFE200F8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode user register 15 | TAUD0CMUR15 | 00 _H | 8 | FFE200FC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 0 | TAUD0CSR0 | 00 _H | 8 | FFE20140 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 1 | TAUD0CSR1 | 00 _H | 8 | FFE20144 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 2 | TAUD0CSR2 | 00 _H | 8 | FFE20148 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 3 | TAUD0CSR3 | 00 _H | 8 | FFE2014C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 4 | TAUD0CSR4 | 00 _H | 8 | FFE20150 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 5 | TAUD0CSR5 | 00 _H | 8 | FFE20154 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 6 | TAUD0CSR6 | 00 _H | 8 | FFE20158 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 7 | TAUD0CSR7 | 00 _H | 8 | FFE2015C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (94/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD0 | TAUD0 channel status register 8 | TAUD0CSR8 | 00 _H | 8 | FFE20160 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 9 | TAUD0CSR9 | 00 _H | 8 | FFE20164 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 10 | TAUD0CSR10 | 00 _H | 8 | FFE20168 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 11 | TAUD0CSR11 | 00 _H | 8 | FFE2016C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 12 | TAUD0CSR12 | 00 _H | 8 | FFE20170 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 13 | TAUD0CSR13 | 00 _H | 8 | FFE20174 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 14 | TAUD0CSR14 | 00 _H | 8 | FFE20178 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel status register 15 | TAUD0CSR15 | 00 _H | 8 | FFE2017C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 0 | TAUD0CSC0 | 00 _H | 8 | FFE20180 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 1 | TAUD0CSC1 | 00 _H | 8 | FFE20184 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 2 | TAUD0CSC2 | 00 _H | 8 | FFE20188 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 3 | TAUD0CSC3 | 00 _H | 8 | FFE2018C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 4 | TAUD0CSC4 | 00 _H | 8 | FFE20190 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 5 | TAUD0CSC5 | 00 _H | 8 | FFE20194 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 6 | TAUD0CSC6 | 00 _H | 8 | FFE20198 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 7 | TAUD0CSC7 | 00 _H | 8 | FFE2019C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 8 | TAUD0CSC8 | 00 _H | 8 | FFE201A0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 9 | TAUD0CSC9 | 00 _H | 8 | FFE201A4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 10 | TAUD0CSC10 | 00 _H | 8 | FFE201A8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 11 | TAUD0CSC11 | 00 _H | 8 | FFE201AC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 12 | TAUD0CSC12 | 00 _H | 8 | FFE201B0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 13 | TAUD0CSC13 | 00 _H | 8 | FFE201B4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 14 | TAUD0CSC14 | 00 _H | 8 | FFE201B8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 Channel Status Clear Register 15 | TAUD0CSC15 | 00 _H | 8 | FFE201BC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel enable status register | TAUD0TE | 0000 _H | 16 | FFE201C0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel start trigger register | TAUD0TS | 0000 _H | 16 | FFE201C4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel stop trigger register | TAUD0TT | 0000 _H | 16 | FFE201C8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 0 | TAUD0CMOR0 | 0000 _H | 16 | FFE20200 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 1 | TAUD0CMOR1 | 0000 _H | 16 | FFE20204 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 2 | TAUD0CMOR2 | 0000 _H | 16 | FFE20208 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 3 | TAUD0CMOR3 | 0000 _H | 16 | FFE2020C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 4 | TAUD0CMOR4 | 0000 _H | 16 | FFE20210 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 5 | TAUD0CMOR5 | 0000 _H | 16 | FFE20214 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 6 | TAUD0CMOR6 | 0000 _H | 16 | FFE20218 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 7 | TAUD0CMOR7 | 0000 _H | 16 | FFE2021C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 8 | TAUD0CMOR8 | 0000 _H | 16 | FFE20220 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 9 | TAUD0CMOR9 | 0000 _H | 16 | FFE20224 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 10 | TAUD0CMOR10 | 0000 _H | 16 | FFE20228 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 11 | TAUD0CMOR11 | 0000 _H | 16 | FFE2022C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 12 | TAUD0CMOR12 | 0000 _H | 16 | FFE20230 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 13 | TAUD0CMOR13 | 0000 _H | 16 | FFE20234 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 14 | TAUD0CMOR14 | 0000 _H | 16 | FFE20238 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel mode OS register 15 | TAUD0CMOR15 | 0000 _H | 16 | FFE2023C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 prescaler clock select register | TAUD0TPS | FFFF _H | 16 | FFE20240 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 prescaler baud rate setting register | TAUD0BRS | 00 _H | 8 | FFE20244 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel output mode register | TAUD0TOM | 0000 _H | 16 | FFE20248 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel output configuration register | TAUD0TOC | 0000 _H | 16 | FFE2024C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel dead time output enable register | TAUD0TDE | 0000 _H | 16 | FFE20250 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (95/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD0 | TAUD0 channel dead time output mode register | TAUD0TDM | 0000 _H | 16 | FFE20254 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel real-time output enable register | TAUD0TRE | 0000 _H | 16 | FFE20258 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel real-time output control register | TAUD0TRC | 0000 _H | 16 | FFE2025C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel reload data enable register | TAUD0RDE | 0000 _H | 16 | FFE20260 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel reload data mode register | TAUD0RDM | 0000 _H | 16 | FFE20264 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel reload data control CH select register | TAUD0RDS | 0000 _H | 16 | FFE20268 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD0 | TAUD0 channel reload data control register | TAUD0RDC | 0000 _H | 16 | FFE2026C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 0 | TAUD1CDR0 | 0000 _H | 16 | FFE21000 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 1 | TAUD1CDR1 | 0000 _H | 16 | FFE21004 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 2 | TAUD1CDR2 | 0000 _H | 16 | FFE21008 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 3 | TAUD1CDR3 | 0000 _H | 16 | FFE2100C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 4 | TAUD1CDR4 | 0000 _H | 16 | FFE21010 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 5 | TAUD1CDR5 | 0000 _H | 16 | FFE21014 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 6 | TAUD1CDR6 | 0000 _H | 16 | FFE21018 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 7 | TAUD1CDR7 | 0000 _H | 16 | FFE2101C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 8 | TAUD1CDR8 | 0000 _H | 16 | FFE21020 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 9 | TAUD1CDR9 | 0000 _H | 16 | FFE21024 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 10 | TAUD1CDR10 | 0000 _H | 16 | FFE21028 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 11 | TAUD1CDR11 | 0000 _H | 16 | FFE2102C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 12 | TAUD1CDR12 | 0000 _H | 16 | FFE21030 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 13 | TAUD1CDR13 | 0000 _H | 16 | FFE21034 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 14 | TAUD1CDR14 | 0000 _H | 16 | FFE21038 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel data register 15 | TAUD1CDR15 | 0000 _H | 16 | FFE2103C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel output active level register | TAUD1TOL | 0000 _H | 16 | FFE21040 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel reload data trigger register | TAUD1RDT | 0000 _H | 16 | FFE21044 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel reload status register | TAUD1RSF | 0000 _H | 16 | FFE21048 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel real-time output register | TAUD1TRO | 0000 _H | 16 | FFE2104C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel modulation output enable register | TAUD1TME | 0000 _H | 16 | FFE21050 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel dead time output level register | TAUD1TDL | 0000 _H | 16 | FFE21054 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel output register | TAUD1TO | 0000 _H | 16 | FFE21058 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel output enable register | TAUD1TOE | 0000 _H | 16 | FFE2105C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 0 | TAUD1CNT0 | FFFF _H | 16 | FFE21080 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 1 | TAUD1CNT1 | FFFF _H | 16 | FFE21084 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 2 | TAUD1CNT2 | FFFF _H | 16 | FFE21088 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 3 | TAUD1CNT3 | FFFF _H | 16 | FFE2108C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 4 | TAUD1CNT4 | FFFF _H | 16 | FFE21090 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 5 | TAUD1CNT5 | FFFF _H | 16 | FFE21094 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 6 | TAUD1CNT6 | FFFF _H | 16 | FFE21098 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 7 | TAUD1CNT7 | FFFF _H | 16 | FFE2109C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 8 | TAUD1CNT8 | FFFF _H | 16 | FFE210A0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 9 | TAUD1CNT9 | FFFF _H | 16 | FFE210A4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 10 | TAUD1CNT10 | FFFF _H | 16 | FFE210A8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 11 | TAUD1CNT11 | FFFF _H | 16 | FFE210AC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 12 | TAUD1CNT12 | FFFF _H | 16 | FFE210B0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 13 | TAUD1CNT13 | FFFF _H | 16 | FFE210B4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 14 | TAUD1CNT14 | FFFF _H | 16 | FFE210B8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel counter register 15 | TAUD1CNT15 | FFFF _H | 16 | FFE210BC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (96/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD1 | TAUD1 channel mode user register 0 | TAUD1CMUR0 | 00 _H | 8 | FFE210C0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 1 | TAUD1CMUR1 | 00 _H | 8 | FFE210C4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 2 | TAUD1CMUR2 | 00 _H | 8 | FFE210C8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 3 | TAUD1CMUR3 | 00 _H | 8 | FFE210CC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 4 | TAUD1CMUR4 | 00 _H | 8 | FFE210D0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 5 | TAUD1CMUR5 | 00 _H | 8 | FFE210D4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 6 | TAUD1CMUR6 | 00 _H | 8 | FFE210D8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 7 | TAUD1CMUR7 | 00 _H | 8 | FFE210DC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 8 | TAUD1CMUR8 | 00 _H | 8 | FFE210E0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 9 | TAUD1CMUR9 | 00 _H | 8 | FFE210E4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 10 | TAUD1CMUR10 | 00 _H | 8 | FFE210E8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 11 | TAUD1CMUR11 | 00 _H | 8 | FFE210EC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 12 | TAUD1CMUR12 | 00 _H | 8 | FFE210F0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 13 | TAUD1CMUR13 | 00 _H | 8 | FFE210F4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 14 | TAUD1CMUR14 | 00 _H | 8 | FFE210F8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode user register 15 | TAUD1CMUR15 | 00 _H | 8 | FFE210FC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 0 | TAUD1CSR0 | 00 _H | 8 | FFE21140 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 1 | TAUD1CSR1 | 00 _H | 8 | FFE21144 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 2 | TAUD1CSR2 | 00 _H | 8 | FFE21148 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 3 | TAUD1CSR3 | 00 _H | 8 | FFE2114C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 4 | TAUD1CSR4 | 00 _H | 8 | FFE21150 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 5 | TAUD1CSR5 | 00 _H | 8 | FFE21154 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 6 | TAUD1CSR6 | 00 _H | 8 | FFE21158 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 7 | TAUD1CSR7 | 00 _H | 8 | FFE2115C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 8 | TAUD1CSR8 | 00 _H | 8 | FFE21160 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 9 | TAUD1CSR9 | 00 _H | 8 | FFE21164 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 10 | TAUD1CSR10 | 00 _H | 8 | FFE21168 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 11 | TAUD1CSR11 | 00 _H | 8 | FFE2116C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 12 | TAUD1CSR12 | 00 _H | 8 | FFE21170 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 13 | TAUD1CSR13 | 00 _H | 8 | FFE21174 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 14 | TAUD1CSR14 | 00 _H | 8 | FFE21178 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel status register 15 | TAUD1CSR15 | 00 _H | 8 | FFE2117C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 0 | TAUD1CSC0 | 00 _H | 8 | FFE21180 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 1 | TAUD1CSC1 | 00 _H | 8 | FFE21184 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 2 | TAUD1CSC2 | 00 _H | 8 | FFE21188 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 3 | TAUD1CSC3 | 00 _H | 8 | FFE2118C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 4 | TAUD1CSC4 | 00 _H | 8 | FFE21190 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 5 | TAUD1CSC5 | 00 _H | 8 | FFE21194 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 6 | TAUD1CSC6 | 00 _H | 8 | FFE21198 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 7 | TAUD1CSC7 | 00 _H | 8 | FFE2119C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 8 | TAUD1CSC8 | 00 _H | 8 | FFE211A0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 9 | TAUD1CSC9 | 00 _H | 8 | FFE211A4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 10 | TAUD1CSC10 | 00 _H | 8 | FFE211A8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 11 | TAUD1CSC11 | 00 _H | 8 | FFE211AC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 12 | TAUD1CSC12 | 00 _H | 8 | FFE211B0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 13 | TAUD1CSC13 | 00 _H | 8 | FFE211B4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 14 | TAUD1CSC14 | 00 _H | 8 | FFE211B8 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 Channel Status Clear Register 15 | TAUD1CSC15 | 00 _H | 8 | FFE211BC _H | 1 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (97/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD1 | TAUD1 channel enable status register | TAUD1TE | 0000 _H | 16 | FFE211C0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel start trigger register | TAUD1TS | 0000 _H | 16 | FFE211C4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel stop trigger register | TAUD1TT | 0000 _H | 16 | FFE211C8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 0 | TAUD1CMOR0 | 0000 _H | 16 | FFE21200 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 1 | TAUD1CMOR1 | 0000 _H | 16 | FFE21204 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 2 | TAUD1CMOR2 | 0000 _H | 16 | FFE21208 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 3 | TAUD1CMOR3 | 0000 _H | 16 | FFE2120C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 4 | TAUD1CMOR4 | 0000 _H | 16 | FFE21210 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 5 | TAUD1CMOR5 | 0000 _H | 16 | FFE21214 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 6 | TAUD1CMOR6 | 0000 _H | 16 | FFE21218 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 7 | TAUD1CMOR7 | 0000 _H | 16 | FFE2121C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 8 | TAUD1CMOR8 | 0000 _H | 16 | FFE21220 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 9 | TAUD1CMOR9 | 0000 _H | 16 | FFE21224 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 10 | TAUD1CMOR10 | 0000 _H | 16 | FFE21228 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 11 | TAUD1CMOR11 | 0000 _H | 16 | FFE2122C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 12 | TAUD1CMOR12 | 0000 _H | 16 | FFE21230 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 13 | TAUD1CMOR13 | 0000 _H | 16 | FFE21234 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 14 | TAUD1CMOR14 | 0000 _H | 16 | FFE21238 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel mode OS register 15 | TAUD1CMOR15 | 0000 _H | 16 | FFE2123C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 prescaler clock select register | TAUD1TPS | FFFF _H | 16 | FFE21240 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 prescaler baud rate setting register | TAUD1BRS | 00 _H | 8 | FFE21244 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel output mode register | TAUD1TOM | 0000 _H | 16 | FFE21248 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel output configuration register | TAUD1TOC | 0000 _H | 16 | FFE2124C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel dead time output enable register | TAUD1TDE | 0000 _H | 16 | FFE21250 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel dead time output mode register | TAUD1TDM | 0000 _H | 16 | FFE21254 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel real-time output enable register | TAUD1TRE | 0000 _H | 16 | FFE21258 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel real-time output control register | TAUD1TRC | 0000 _H | 16 | FFE2125C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel reload data enable register | TAUD1RDE | 0000 _H | 16 | FFE21260 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel reload data mode register | TAUD1RDM | 0000 _H | 16 | FFE21264 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel reload data control CH select register | TAUD1RDS | 0000 _H | 16 | FFE21268 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD1 | TAUD1 channel reload data control register | TAUD1RDC | 0000 _H | 16 | FFE2126C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 0 | TAUD2CDR0 | 0000 _H | 16 | FFE22000 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 1 | TAUD2CDR1 | 0000 _H | 16 | FFE22004 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 2 | TAUD2CDR2 | 0000 _H | 16 | FFE22008 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 3 | TAUD2CDR3 | 0000 _H | 16 | FFE2200C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 4 | TAUD2CDR4 | 0000 _H | 16 | FFE22010 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 5 | TAUD2CDR5 | 0000 _H | 16 | FFE22014 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 6 | TAUD2CDR6 | 0000 _H | 16 | FFE22018 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 7 | TAUD2CDR7 | 0000 _H | 16 | FFE2201C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 8 | TAUD2CDR8 | 0000 _H | 16 | FFE22020 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 9 | TAUD2CDR9 | 0000 _H | 16 | FFE22024 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 10 | TAUD2CDR10 | 0000 _H | 16 | FFE22028 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 11 | TAUD2CDR11 | 0000 _H | 16 | FFE2202C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 12 | TAUD2CDR12 | 0000 _H | 16 | FFE22030 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 13 | TAUD2CDR13 | 0000 _H | 16 | FFE22034 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 14 | TAUD2CDR14 | 0000 _H | 16 | FFE22038 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUD2 | TAUD2 channel data register 15 | TAUD2CDR15 | 0000 _H | 16 | FFE2203C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (98/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD2 | TAUD2 channel output active level register | TAUD2TOL | 0000 _H | 16 | FFE22040 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel reload data trigger register | TAUD2RDT | 0000 _H | 16 | FFE22044 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel reload status register | TAUD2RSF | 0000 _H | 16 | FFE22048 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel real-time output register | TAUD2TRO | 0000 _H | 16 | FFE2204C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel modulation output enable register | TAUD2TME | 0000 _H | 16 | FFE22050 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel dead time output level register | TAUD2TDL | 0000 _H | 16 | FFE22054 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel output register | TAUD2TO | 0000 _H | 16 | FFE22058 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel output enable register | TAUD2TOE | 0000 _H | 16 | FFE2205C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 0 | TAUD2CNT0 | FFFF _H | 16 | FFE22080 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 1 | TAUD2CNT1 | FFFF _H | 16 | FFE22084 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 2 | TAUD2CNT2 | FFFF _H | 16 | FFE22088 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 3 | TAUD2CNT3 | FFFF _H | 16 | FFE2208C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 4 | TAUD2CNT4 | FFFF _H | 16 | FFE22090 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 5 | TAUD2CNT5 | FFFF _H | 16 | FFE22094 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 6 | TAUD2CNT6 | FFFF _H | 16 | FFE22098 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 7 | TAUD2CNT7 | FFFF _H | 16 | FFE2209C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 8 | TAUD2CNT8 | FFFF _H | 16 | FFE220A0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 9 | TAUD2CNT9 | FFFF _H | 16 | FFE220A4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 10 | TAUD2CNT10 | FFFF _H | 16 | FFE220A8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 11 | TAUD2CNT11 | FFFF _H | 16 | FFE220AC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 12 | TAUD2CNT12 | FFFF _H | 16 | FFE220B0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 13 | TAUD2CNT13 | FFFF _H | 16 | FFE220B4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 14 | TAUD2CNT14 | FFFF _H | 16 | FFE220B8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel counter register 15 | TAUD2CNT15 | FFFF _H | 16 | FFE220BC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 0 | TAUD2CMUR0 | 00 _H | 8 | FFE220C0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 1 | TAUD2CMUR1 | 00 _H | 8 | FFE220C4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 2 | TAUD2CMUR2 | 00 _H | 8 | FFE220C8 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 3 | TAUD2CMUR3 | 00 _H | 8 | FFE220CC _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 4 | TAUD2CMUR4 | 00 _H | 8 | FFE220D0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 5 | TAUD2CMUR5 | 00 _H | 8 | FFE220D4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 6 | TAUD2CMUR6 | 00 _H | 8 | FFE220D8 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 7 | TAUD2CMUR7 | 00 _H | 8 | FFE220DC _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 8 | TAUD2CMUR8 | 00 _H | 8 | FFE220E0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 9 | TAUD2CMUR9 | 00 _H | 8 | FFE220E4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 10 | TAUD2CMUR10 | 00 _H | 8 | FFE220E8 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 11 | TAUD2CMUR11 | 00 _H | 8 | FFE220EC _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 12 | TAUD2CMUR12 | 00 _H | 8 | FFE220F0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 13 | TAUD2CMUR13 | 00 _H | 8 | FFE220F4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 14 | TAUD2CMUR14 | 00 _H | 8 | FFE220F8 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode user register 15 | TAUD2CMUR15 | 00 _H | 8 | FFE220FC _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 0 | TAUD2CSR0 | 00 _H | 8 | FFE22140 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 1 | TAUD2CSR1 | 00 _H | 8 | FFE22144 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 2 | TAUD2CSR2 | 00 _H | 8 | FFE22148 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 3 | TAUD2CSR3 | 00 _H | 8 | FFE2214C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 4 | TAUD2CSR4 | 00 _H | 8 | FFE22150 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 5 | TAUD2CSR5 | 00 _H | 8 | FFE22154 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 6 | TAUD2CSR6 | 00 _H | 8 | FFE22158 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 7 | TAUD2CSR7 | 00 _H | 8 | FFE2215C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (99/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD2 | TAUD2 channel status register 8 | TAUD2CSR8 | 00 _H | 8 | FFE22160 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 9 | TAUD2CSR9 | 00 _H | 8 | FFE22164 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 10 | TAUD2CSR10 | 00 _H | 8 | FFE22168 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 11 | TAUD2CSR11 | 00 _H | 8 | FFE2216C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 12 | TAUD2CSR12 | 00 _H | 8 | FFE22170 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 13 | TAUD2CSR13 | 00 _H | 8 | FFE22174 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 14 | TAUD2CSR14 | 00 _H | 8 | FFE22178 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel status register 15 | TAUD2CSR15 | 00 _H | 8 | FFE2217C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 0 | TAUD2CSC0 | 00 _H | 8 | FFE22180 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 1 | TAUD2CSC1 | 00 _H | 8 | FFE22184 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 2 | TAUD2CSC2 | 00 _H | 8 | FFE22188 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 3 | TAUD2CSC3 | 00 _H | 8 | FFE2218C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 4 | TAUD2CSC4 | 00 _H | 8 | FFE22190 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 5 | TAUD2CSC5 | 00 _H | 8 | FFE22194 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 6 | TAUD2CSC6 | 00 _H | 8 | FFE22198 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 7 | TAUD2CSC7 | 00 _H | 8 | FFE2219C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 8 | TAUD2CSC8 | 00 _H | 8 | FFE221A0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 9 | TAUD2CSC9 | 00 _H | 8 | FFE221A4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 10 | TAUD2CSC10 | 00 _H | 8 | FFE221A8 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 11 | TAUD2CSC11 | 00 _H | 8 | FFE221AC _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 12 | TAUD2CSC12 | 00 _H | 8 | FFE221B0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 13 | TAUD2CSC13 | 00 _H | 8 | FFE221B4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 14 | TAUD2CSC14 | 00 _H | 8 | FFE221B8 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 Channel Status Clear Register 15 | TAUD2CSC15 | 00 _H | 8 | FFE221BC _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel enable status register | TAUD2TE | 0000 _H | 16 | FFE221C0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel start trigger register | TAUD2TS | 0000 _H | 16 | FFE221C4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel stop trigger register | TAUD2TT | 0000 _H | 16 | FFE221C8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 0 | TAUD2CMOR0 | 0000 _H | 16 | FFE22200 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 1 | TAUD2CMOR1 | 0000 _H | 16 | FFE22204 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 2 | TAUD2CMOR2 | 0000 _H | 16 | FFE22208 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 3 | TAUD2CMOR3 | 0000 _H | 16 | FFE2220C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 4 | TAUD2CMOR4 | 0000 _H | 16 | FFE22210 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 5 | TAUD2CMOR5 | 0000 _H | 16 | FFE22214 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 6 | TAUD2CMOR6 | 0000 _H | 16 | FFE22218 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 7 | TAUD2CMOR7 | 0000 _H | 16 | FFE2221C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 8 | TAUD2CMOR8 | 0000 _H | 16 | FFE22220 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 9 | TAUD2CMOR9 | 0000 _H | 16 | FFE22224 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 10 | TAUD2CMOR10 | 0000 _H | 16 | FFE22228 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 11 | TAUD2CMOR11 | 0000 _H | 16 | FFE2222C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 12 | TAUD2CMOR12 | 0000 _H | 16 | FFE22230 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 13 | TAUD2CMOR13 | 0000 _H | 16 | FFE22234 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 14 | TAUD2CMOR14 | 0000 _H | 16 | FFE22238 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel mode OS register 15 | TAUD2CMOR15 | 0000 _H | 16 | FFE2223C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 prescaler clock select register | TAUD2TPS | FFFF _H | 16 | FFE22240 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 prescaler baud rate setting register | TAUD2BRS | 00 _H | 8 | FFE22244 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel output mode register | TAUD2TOM | 0000 _H | 16 | FFE22248 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel output configuration register | TAUD2TOC | 0000 _H | 16 | FFE2224C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel dead time output enable register | TAUD2TDE | 0000 _H | 16 | FFE22250 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (100/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUD2 | TAUD2 channel dead time output mode register | TAUD2TDM | 0000 _H | 16 | FFE22254 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel real-time output enable register | TAUD2TRE | 0000 _H | 16 | FFE22258 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel real-time output control register | TAUD2TRC | 0000 _H | 16 | FFE2225C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel reload data enable register | TAUD2RDE | 0000 _H | 16 | FFE22260 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel reload data mode register | TAUD2RDM | 0000 _H | 16 | FFE22264 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel reload data control CH select register | TAUD2RDS | 0000 _H | 16 | FFE22268 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUD2 | TAUD2 channel reload data control register | TAUD2RDC | 0000 _H | 16 | FFE2226C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel data register 0 | TAUJ0CDR0 | 00000000 _H | 32 | FFE50000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel data register 1 | TAUJ0CDR1 | 00000000 _H | 32 | FFE50004 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel data register 2 | TAUJ0CDR2 | 00000000 _H | 32 | FFE50008 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel data register 3 | TAUJ0CDR3 | 00000000 _H | 32 | FFE5000C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel counter register 0 | TAUJ0CNT0 | FFFFFFFF _H | 32 | FFE50010 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel counter register 1 | TAUJ0CNT1 | FFFFFFFF _H | 32 | FFE50014 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel counter register 2 | TAUJ0CNT2 | FFFFFFFF _H | 32 | FFE50018 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel counter register 3 | TAUJ0CNT3 | FFFFFFFF _H | 32 | FFE5001C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode user register 0 | TAUJ0CMUR0 | 00 _H | 8 | FFE50020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode user register 1 | TAUJ0CMUR1 | 00 _H | 8 | FFE50024 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode user register 2 | TAUJ0CMUR2 | 00 _H | 8 | FFE50028 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode user register 3 | TAUJ0CMUR3 | 00 _H | 8 | FFE5002C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel status register 0 | TAUJ0CSR0 | 00 _H | 8 | FFE50030 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel status register 1 | TAUJ0CSR1 | 00 _H | 8 | FFE50034 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel status register 2 | TAUJ0CSR2 | 00 _H | 8 | FFE50038 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel status register 3 | TAUJ0CSR3 | 00 _H | 8 | FFE5003C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 Channel Status Clear Register 0 | TAUJ0CSC0 | 00 _H | 8 | FFE50040 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 Channel Status Clear Register 1 | TAUJ0CSC1 | 00 _H | 8 | FFE50044 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 Channel Status Clear Register 2 | TAUJ0CSC2 | 00 _H | 8 | FFE50048 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 Channel Status Clear Register 3 | TAUJ0CSC3 | 00 _H | 8 | FFE5004C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel enable status register | TAUJ0TE | 00 _H | 8 | FFE50050 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel start trigger register | TAUJ0TS | 00 _H | 8 | FFE50054 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel stop trigger register | TAUJ0TT | 00 _H | 8 | FFE50058 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel output register | TAUJ0TO | 00 _H | 8 | FFE5005C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel output enable register | TAUJ0TOE | 00 _H | 8 | FFE50060 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel output active level register | TAUJ0TOL | 00 _H | 8 | FFE50064 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel reload data trigger register | TAUJ0RDT | 00 _H | 8 | FFE50068 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel reload status register | TAUJ0RSF | 00 _H | 8 | FFE5006C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode OS register 0 | TAUJ0CMOR0 | 0000 _H | 16 | FFE50080 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode OS register 1 | TAUJ0CMOR1 | 0000 _H | 16 | FFE50084 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode OS register 2 | TAUJ0CMOR2 | 0000 _H | 16 | FFE50088 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel mode OS register 3 | TAUJ0CMOR3 | 0000 _H | 16 | FFE5008C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 prescaler clock select register | TAUJ0TPS | FFFF _H | 16 | FFE50090 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 prescaler baud rate setting register | TAUJ0BRS | 00 _H | 8 | FFE50094 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel output mode register | TAUJ0TOM | 00 _H | 8 | FFE50098 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel output configuration register | TAUJ0TOC | 00 _H | 8 | FFE5009C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel reload data enable register | TAUJ0RDE | 00 _H | 8 | FFE500A0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ0 | TAUJ0 channel reload data mode register | TAUJ0RDM | 00 _H | 8 | FFE500A4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ1 | TAUJ1 channel data register 0 | TAUJ1CDR0 | 00000000 _H | 32 | FFE51000 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ1 | TAUJ1 channel data register 1 | TAUJ1CDR1 | 00000000 _H | 32 | FFE51004 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (101/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUJ1 | TAUJ1 channel data register 2 | TAUJ1CDR2 | 00000000 _H | 32 | FFE51008 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel data register 3 | TAUJ1CDR3 | 00000000 _H | 32 | FFE5100C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel counter register 0 | TAUJ1CNT0 | FFFFFFFF _H | 32 | FFE51010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel counter register 1 | TAUJ1CNT1 | FFFFFFFF _H | 32 | FFE51014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel counter register 2 | TAUJ1CNT2 | FFFFFFFF _H | 32 | FFE51018 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel counter register 3 | TAUJ1CNT3 | FFFFFFFF _H | 32 | FFE5101C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode user register 0 | TAUJ1CMUR0 | 00 _H | 8 | FFE51020 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode user register 1 | TAUJ1CMUR1 | 00 _H | 8 | FFE51024 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode user register 2 | TAUJ1CMUR2 | 00 _H | 8 | FFE51028 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode user register 3 | TAUJ1CMUR3 | 00 _H | 8 | FFE5102C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel status register 0 | TAUJ1CSR0 | 00 _H | 8 | FFE51030 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel status register 1 | TAUJ1CSR1 | 00 _H | 8 | FFE51034 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel status register 2 | TAUJ1CSR2 | 00 _H | 8 | FFE51038 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel status register 3 | TAUJ1CSR3 | 00 _H | 8 | FFE5103C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 Channel Status Clear Register 0 | TAUJ1CSC0 | 00 _H | 8 | FFE51040 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 Channel Status Clear Register 1 | TAUJ1CSC1 | 00 _H | 8 | FFE51044 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 Channel Status Clear Register 2 | TAUJ1CSC2 | 00 _H | 8 | FFE51048 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 Channel Status Clear Register 3 | TAUJ1CSC3 | 00 _H | 8 | FFE5104C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel enable status register | TAUJ1TE | 00 _H | 8 | FFE51050 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel start trigger register | TAUJ1TS | 00 _H | 8 | FFE51054 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel stop trigger register | TAUJ1TT | 00 _H | 8 | FFE51058 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel output register | TAUJ1TO | 00 _H | 8 | FFE5105C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel output enable register | TAUJ1TOE | 00 _H | 8 | FFE51060 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel output active level register | TAUJ1TOL | 00 _H | 8 | FFE51064 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel reload data trigger register | TAUJ1RDT | 00 _H | 8 | FFE51068 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel reload status register | TAUJ1RSF | 00 _H | 8 | FFE5106C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode OS register 0 | TAUJ1CMOR0 | 0000 _H | 16 | FFE51080 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode OS register 1 | TAUJ1CMOR1 | 0000 _H | 16 | FFE51084 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode OS register 2 | TAUJ1CMOR2 | 0000 _H | 16 | FFE51088 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel mode OS register 3 | TAUJ1CMOR3 | 0000 _H | 16 | FFE5108C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 prescaler clock select register | TAUJ1TPS | FFFF _H | 16 | FFE51090 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 prescaler baud rate setting register | TAUJ1BRS | 00 _H | 8 | FFE51094 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel output mode register | TAUJ1TOM | 00 _H | 8 | FFE51098 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel output configuration register | TAUJ1TOC | 00 _H | 8 | FFE5109C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel reload data enable register | TAUJ1RDE | 00 _H | 8 | FFE510A0 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ1 | TAUJ1 channel reload data mode register | TAUJ1RDM | 00 _H | 8 | FFE510A4 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel data register 0 | TAUJ2CDR0 | 00000000 _H | 32 | FFE52000 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel data register 1 | TAUJ2CDR1 | 00000000 _H | 32 | FFE52004 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel data register 2 | TAUJ2CDR2 | 00000000 _H | 32 | FFE52008 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel data register 3 | TAUJ2CDR3 | 00000000 _H | 32 | FFE5200C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel counter register 0 | TAUJ2CNT0 | FFFFFFFF _H | 32 | FFE52010 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel counter register 1 | TAUJ2CNT1 | FFFFFFFF _H | 32 | FFE52014 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel counter register 2 | TAUJ2CNT2 | FFFFFFFF _H | 32 | FFE52018 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel counter register 3 | TAUJ2CNT3 | FFFFFFFF _H | 32 | FFE5201C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel mode user register 0 | TAUJ2CMUR0 | 00 _H | 8 | FFE52020 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel mode user register 1 | TAUJ2CMUR1 | 00 _H | 8 | FFE52024 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel mode user register 2 | TAUJ2CMUR2 | 00 _H | 8 | FFE52028 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TAUJ2 | TAUJ2 channel mode user register 3 | TAUJ2CMUR3 | 00 _H | 8 | FFE5202C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (102/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAUJ2 | TAUJ2 channel status register 0 | TAUJ2CSR0 | 00 _H | 8 | FFE52030 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel status register 1 | TAUJ2CSR1 | 00 _H | 8 | FFE52034 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel status register 2 | TAUJ2CSR2 | 00 _H | 8 | FFE52038 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel status register 3 | TAUJ2CSR3 | 00 _H | 8 | FFE5203C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 Channel Status Clear Register 0 | TAUJ2CSC0 | 00 _H | 8 | FFE52040 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 Channel Status Clear Register 1 | TAUJ2CSC1 | 00 _H | 8 | FFE52044 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 Channel Status Clear Register 2 | TAUJ2CSC2 | 00 _H | 8 | FFE52048 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 Channel Status Clear Register 3 | TAUJ2CSC3 | 00 _H | 8 | FFE5204C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel enable status register | TAUJ2TE | 00 _H | 8 | FFE52050 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel start trigger register | TAUJ2TS | 00 _H | 8 | FFE52054 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel stop trigger register | TAUJ2TT | 00 _H | 8 | FFE52058 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel output register | TAUJ2TO | 00 _H | 8 | FFE5205C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel output enable register | TAUJ2TOE | 00 _H | 8 | FFE52060 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel output active level register | TAUJ2TOL | 00 _H | 8 | FFE52064 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel reload data trigger register | TAUJ2RDT | 00 _H | 8 | FFE52068 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel reload status register | TAUJ2RSF | 00 _H | 8 | FFE5206C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel mode OS register 0 | TAUJ2CMOR0 | 0000 _H | 16 | FFE52080 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel mode OS register 1 | TAUJ2CMOR1 | 0000 _H | 16 | FFE52084 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel mode OS register 2 | TAUJ2CMOR2 | 0000 _H | 16 | FFE52088 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel mode OS register 3 | TAUJ2CMOR3 | 0000 _H | 16 | FFE5208C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 prescaler clock select register | TAUJ2TPS | FFFF _H | 16 | FFE52090 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 prescaler baud rate setting register | TAUJ2BRS | 00 _H | 8 | FFE52094 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel output mode register | TAUJ2TOM | 00 _H | 8 | FFE52098 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel output configuration register | TAUJ2TOC | 00 _H | 8 | FFE5209C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel reload data enable register | TAUJ2RDE | 00 _H | 8 | FFE520A0 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAUJ2 | TAUJ2 channel reload data mode register | TAUJ2RDM | 00 _H | 8 | FFE520A4 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 I/O control register 2 | TSG30IOC2 | 0000 _H | 16 | FFE70000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 control register 3 | TSG30CTL3 | 00 _H | 8 | FFE70004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 control register 5 | TSG30CTL5 | 0000 _H | 16 | FFE70008 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 control register 6 | TSG30CTL6 | 0000 _H | 16 | FFE7000C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 status register 0 | TSG30STR0 | 00 _H | 8 | FFE70010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 status register 1 | TSG30STR1 | 00 _H | 8 | FFE70014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 status register 2 | TSG30STR2 | 0000 _H | 16 | FFE70018 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 status clear trigger register | TSG30STC | 0000 _H | 16 | FFE7001C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 option register 0 | TSG30OPT0 | 00 _H | 8 | FFE70020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 option register 1 | TSG30OPT1 | 00 _H | 8 | FFE70024 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 counter read buffer register | TSG30CNT | 0000 _H | 16 | FFE70028 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 sub-counter read buffer register | TSG30SBC | 0000 _H | 16 | FFE7002C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 trigger register 0 | TSG30TRG0 | 00 _H | 8 | FFE70030 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 trigger register 1 | TSG30TRG1 | 00 _H | 8 | FFE70034 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 trigger register 2 | TSG30TRG2 | 00 _H | 8 | FFE70038 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 compare register 1, 2 | TSG30CMP1W | 00000000 _H | 32 | FFE70040 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 compare register 5, 6 | TSG30CMP5W | 00000000 _H | 32 | FFE70044 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 compare register 9, 10 | TSG30CMP9W | 00000000 _H | 32 | FFE70048 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 compare register 3, 4 | TSG30CMP3W | 00000000 _H | 32 | FFE7004C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 compare register 7, 8 | TSG30CMP7W | 00000000 _H | 32 | FFE70050 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 compare register 11, 12 | TSG30CMP11W | 00000000 _H | 32 | FFE70054 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 compare register 0 | TSG30CMP0 | 00000000 _H | 32 | FFE70058 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (103/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TSG30 | TSG30 diagnostic output compare register 0, 1 | TSG30DCMP0W | 00000000 _H | 32 | FFE7005C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 diagnostic output compare register 2 | TSG30DCMP2 | 00000000 _H | 32 | FFE70060 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 pattern register 0 | TSG30PAT0W | 00000000 _H | 32 | FFE70064 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 pattern register 1 | TSG30PAT1W | 00000000 _H | 32 | FFE70068 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 dead time control register 0 | TSG30DTC0W | 00000000 _H | 32 | FFE7006C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 dead time control register 1 | TSG30DTC1W | 00000000 _H | 32 | FFE70070 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 I/O control register 3 | TSG30IOC3 | 00000000 _H | 32 | FFE70074 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 control register 2 | TSG30CTL2 | 00000000 _H | 32 | FFE70078 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 control register 4 | TSG30CTL4 | 00000000 _H | 32 | FFE7007C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 1 | TSG30CMP1 | 0000 _H | 16 | FFE70080 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 2 | TSG30CMP2 | 0000 _H | 16 | FFE70084 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 5 | TSG30CMP5 | 0000 _H | 16 | FFE70088 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 6 | TSG30CMP6 | 0000 _H | 16 | FFE7008C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 9 | TSG30CMP9 | 0000 _H | 16 | FFE70090 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 10 | TSG30CMP10 | 0000 _H | 16 | FFE70094 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 3 | TSG30CMP3 | 0000 _H | 16 | FFE70098 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 4 | TSG30CMP4 | 0000 _H | 16 | FFE7009C _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 7 | TSG30CMP7 | 0000 _H | 16 | FFE700A0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 8 | TSG30CMP8 | 0000 _H | 16 | FFE700A4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 11 | TSG30CMP11 | 0000 _H | 16 | FFE700A8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 compare register 12 | TSG30CMP12 | 0000 _H | 16 | FFE700AC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HT-PWM U phase compare register | TSG30CMPU | 0000 _H | 16 | FFE700B0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HT-PWM V phase compare register | TSG30CMPV | 0000 _H | 16 | FFE700B4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HT-PWM W phase compare register | TSG30CMPW | 0000 _H | 16 | FFE700B8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 SP-PWM U phase active width register | TSG30UPW | 0000 _H | 16 | FFE700BC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 SP-PWM V phase active width register | TSG30VPW | 0000 _H | 16 | FFE700C0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 SP-PWM W phase active width register | TSG30WPW | 0000 _H | 16 | FFE700C4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HSP-PWM Mode W phase shift register | TSG30HSPSHWE | 00000000 _H | 32 | FFE70120 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HSP-PWM Mode V phase shift register | TSG30HSPSHVE | 00000000 _H | 32 | FFE70124 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HSP-PWM Mode U phase shift register | TSG30HSPSHUE | 00000000 _H | 32 | FFE70128 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HSP-PWM Mode W Phase Compare Register | TSG30HSPCMWE | 00000000 _H | 32 | FFE7012C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HSP-PWM Mode V Phase Compare Register | TSG30HSPCMVE | 00000000 _H | 32 | FFE70130 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 HSP-PWM Mode U Phase Compare Register | TSG30HSPCMUE | 00000000 _H | 32 | FFE70134 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended diagnostic output compare register 2 | TSG30DCMP2E | 00000000 _H | 32 | FFE70140 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended diagnostic output compare register 1 | TSG30DCMP1E | 00000000 _H | 32 | FFE70144 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended diagnostic output compare register 0 | TSG30DCMP0E | 00000000 _H | 32 | FFE70148 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 0 | TSG30CMP0E | 00000000 _H | 32 | FFE7014C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 12 | TSG30CMP12E | 00000000 _H | 32 | FFE70150 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 11 | TSG30CMP11E | 00000000 _H | 32 | FFE70154 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 8 | TSG30CMP8E | 00000000 _H | 32 | FFE70158 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 7 | TSG30CMP7E | 00000000 _H | 32 | FFE7015C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 4 | TSG30CMP4E | 00000000 _H | 32 | FFE70160 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 3 | TSG30CMP3E | 00000000 _H | 32 | FFE70164 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 10 | TSG30CMP10E | 00000000 _H | 32 | FFE70168 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 9 | TSG30CMP9E | 00000000 _H | 32 | FFE7016C _H | 1 | 32 | √ | √ | √ | √ | √ | √ |
| TSG30 | TSG30 bit extended compare register 6 | TSG30CMP6E | 00000000 _H | 32 | FFE70170 _H | 1 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (104/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TSG30 | TSG30 bit extended compare register 5 | TSG30CMP5E | 00000000 _H | 32 | FFE70174 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended compare register 2 | TSG30CMP2E | 00000000 _H | 32 | FFE70178 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended compare register 1 | TSG30CMP1E | 00000000 _H | 32 | FFE7017C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended HT-PWM W phase compare register | TSG30CMPWE | 00000000 _H | 32 | FFE70180 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended HT-PWM V phase compare register | TSG30CMPVE | 00000000 _H | 32 | FFE70184 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended HT-PWM U phase compare register | TSG30CMPUE | 00000000 _H | 32 | FFE70188 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended SP-PWM W phase active width register | TSG30WPWE | 00000000 _H | 32 | FFE70190 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended SP-PWM V phase active width register | TSG30VPWE | 00000000 _H | 32 | FFE70194 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended SP-PWM U phase active width register | TSG30UPWE | 00000000 _H | 32 | FFE70198 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended counter read buffer register | TSG30CNTE | 00000000 _H | 32 | FFE701A0 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 bit extended sub-counter read buffer register | TSG30SBCE | 00000000 _H | 32 | FFE701A4 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 I/O control register 0 | TSG30IOC0 | 7E _H | 8 | FFE70200 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 I/O control register 1 | TSG30IOC1 | 00 _H | 8 | FFE70204 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 control register 0 | TSG30CTL0 | 00 _H | 8 | FFE70208 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 control register 1 | TSG30CTL1 | 0000 _H | 16 | FFE7020C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 dead time protection register | TSG30DTPR | 0000 _H | 16 | FFE70210 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 control register 7 | TSG30CTL7 | 00 _H | 8 | FFE70218 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG30 | TSG30 control register 8 | TSG30CTL8 | 00 _H | 8 | FFE7021C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 I/O control register 2 | TSG31IOC2 | 0000 _H | 16 | FFE71000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 3 | TSG31CTL3 | 00 _H | 8 | FFE71004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 5 | TSG31CTL5 | 0000 _H | 16 | FFE71008 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 6 | TSG31CTL6 | 0000 _H | 16 | FFE7100C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 status register 0 | TSG31STR0 | 00 _H | 8 | FFE71010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 status register 1 | TSG31STR1 | 00 _H | 8 | FFE71014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 status register 2 | TSG31STR2 | 0000 _H | 16 | FFE71018 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 status clear trigger register | TSG31STC | 0000 _H | 16 | FFE7101C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 option register 0 | TSG31OPT0 | 00 _H | 8 | FFE71020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 option register 1 | TSG31OPT1 | 00 _H | 8 | FFE71024 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 counter read buffer register | TSG31CNT | 0000 _H | 16 | FFE71028 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 sub-counter read buffer register | TSG31SBC | 0000 _H | 16 | FFE7102C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 trigger register 0 | TSG31TRG0 | 00 _H | 8 | FFE71030 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 trigger register 1 | TSG31TRG1 | 00 _H | 8 | FFE71034 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 trigger register 2 | TSG31TRG2 | 00 _H | 8 | FFE71038 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 1, 2 | TSG31CMP1W | 00000000 _H | 32 | FFE71040 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 5, 6 | TSG31CMP5W | 00000000 _H | 32 | FFE71044 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 9, 10 | TSG31CMP9W | 00000000 _H | 32 | FFE71048 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 3, 4 | TSG31CMP3W | 00000000 _H | 32 | FFE7104C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 7, 8 | TSG31CMP7W | 00000000 _H | 32 | FFE71050 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 11, 12 | TSG31CMP11W | 00000000 _H | 32 | FFE71054 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 0 | TSG31CMP0 | 00000000 _H | 32 | FFE71058 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 diagnostic output compare register 0, 1 | TSG31DCMP0W | 00000000 _H | 32 | FFE7105C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 diagnostic output compare register 2 | TSG31DCMP2 | 00000000 _H | 32 | FFE71060 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 pattern register 0 | TSG31PAT0W | 00000000 _H | 32 | FFE71064 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 pattern register 1 | TSG31PAT1W | 00000000 _H | 32 | FFE71068 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 dead time control register 0 | TSG31DTC0W | 00000000 _H | 32 | FFE7106C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (105/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TSG31 | TSG31 dead time control register 1 | TSG31DTC1W | 00000000 _H | 32 | FFE71070 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 I/O control register 3 | TSG31IOC3 | 00000000 _H | 32 | FFE71074 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 2 | TSG31CTL2 | 00000000 _H | 32 | FFE71078 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 4 | TSG31CTL4 | 00000000 _H | 32 | FFE7107C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 1 | TSG31CMP1 | 0000 _H | 16 | FFE71080 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 2 | TSG31CMP2 | 0000 _H | 16 | FFE71084 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 5 | TSG31CMP5 | 0000 _H | 16 | FFE71088 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 6 | TSG31CMP6 | 0000 _H | 16 | FFE7108C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 9 | TSG31CMP9 | 0000 _H | 16 | FFE71090 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 10 | TSG31CMP10 | 0000 _H | 16 | FFE71094 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 3 | TSG31CMP3 | 0000 _H | 16 | FFE71098 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 4 | TSG31CMP4 | 0000 _H | 16 | FFE7109C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 7 | TSG31CMP7 | 0000 _H | 16 | FFE710A0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 8 | TSG31CMP8 | 0000 _H | 16 | FFE710A4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 11 | TSG31CMP11 | 0000 _H | 16 | FFE710A8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 compare register 12 | TSG31CMP12 | 0000 _H | 16 | FFE710AC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HT-PWM U phase compare register | TSG31CMPU | 0000 _H | 16 | FFE710B0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HT-PWM V phase compare register | TSG31CMPV | 0000 _H | 16 | FFE710B4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HT-PWM W phase compare register | TSG31CMPW | 0000 _H | 16 | FFE710B8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 SP-PWM U phase active width register | TSG31UPW | 0000 _H | 16 | FFE710BC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 SP-PWM V phase active width register | TSG31VPW | 0000 _H | 16 | FFE710C0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 SP-PWM W phase active width register | TSG31WPW | 0000 _H | 16 | FFE710C4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HSP-PWM Mode W phase shift register | TSG31HSPSHWE | 00000000 _H | 32 | FFE71120 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HSP-PWM Mode V phase shift register | TSG31HSPSHVE | 00000000 _H | 32 | FFE71124 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HSP-PWM Mode U phase shift register | TSG31HSPSHUE | 00000000 _H | 32 | FFE71128 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HSP-PWM Mode W Phase Compare Register | TSG31HSPCMWE | 00000000 _H | 32 | FFE7112C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HSP-PWM Mode V Phase Compare Register | TSG31HSPCMVE | 00000000 _H | 32 | FFE71130 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 HSP-PWM Mode U Phase Compare Register | TSG31HSPCMUE | 00000000 _H | 32 | FFE71134 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended diagnostic output compare register 2 | TSG31DCMP2E | 00000000 _H | 32 | FFE71140 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended diagnostic output compare register 1 | TSG31DCMP1E | 00000000 _H | 32 | FFE71144 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended diagnostic output compare register 0 | TSG31DCMP0E | 00000000 _H | 32 | FFE71148 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 0 | TSG31CMP0E | 00000000 _H | 32 | FFE7114C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 12 | TSG31CMP12E | 00000000 _H | 32 | FFE71150 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 11 | TSG31CMP11E | 00000000 _H | 32 | FFE71154 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 8 | TSG31CMP8E | 00000000 _H | 32 | FFE71158 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 7 | TSG31CMP7E | 00000000 _H | 32 | FFE7115C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 4 | TSG31CMP4E | 00000000 _H | 32 | FFE71160 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 3 | TSG31CMP3E | 00000000 _H | 32 | FFE71164 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 10 | TSG31CMP10E | 00000000 _H | 32 | FFE71168 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 9 | TSG31CMP9E | 00000000 _H | 32 | FFE7116C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 6 | TSG31CMP6E | 00000000 _H | 32 | FFE71170 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 5 | TSG31CMP5E | 00000000 _H | 32 | FFE71174 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 2 | TSG31CMP2E | 00000000 _H | 32 | FFE71178 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended compare register 1 | TSG31CMP1E | 00000000 _H | 32 | FFE7117C _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended HT-PWM W phase compare register | TSG31CMPWE | 00000000 _H | 32 | FFE71180 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (106/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TSG31 | TSG31 bit extended HT-PWM V phase compare register | TSG31CMPVE | 00000000 _H | 32 | FFE71184 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended HT-PWM U phase compare register | TSG31CMPUE | 00000000 _H | 32 | FFE71188 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended SP-PWM W phase active width register | TSG31WPWE | 00000000 _H | 32 | FFE71190 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended SP-PWM V phase active width register | TSG31VPWE | 00000000 _H | 32 | FFE71194 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended SP-PWM U phase active width register | TSG31UPWE | 00000000 _H | 32 | FFE71198 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended counter read buffer register | TSG31CNTE | 00000000 _H | 32 | FFE711A0 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 bit extended sub-counter read buffer register | TSG31SBCE | 00000000 _H | 32 | FFE711A4 _H | 1 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 I/O control register 0 | TSG31IOC0 | 7E _H | 8 | FFE71200 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 I/O control register 1 | TSG31IOC1 | 00 _H | 8 | FFE71204 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 0 | TSG31CTL0 | 00 _H | 8 | FFE71208 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 1 | TSG31CTL1 | 0000 _H | 16 | FFE7120C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 dead time protection register | TSG31DTPR | 0000 _H | 16 | FFE71210 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 7 | TSG31CTL7 | 00 _H | 8 | FFE71218 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TSG31 | TSG31 control register 8 | TSG31CTL8 | 00 _H | 8 | FFE7121C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA capture compare register 0 | ENCA0CCR0 | 0000 _H | 16 | FFE80000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA capture compare register 1 | ENCA0CCR1 | 0000 _H | 16 | FFE80004 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA counter register | ENCA0CNT | 0000 _H | 16 | FFE80008 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA status flag register | ENCA0FLG | 00 _H | 8 | FFE8000C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA status flag clear register | ENCA0FGC | 00 _H | 8 | FFE80010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA timer enable status register | ENCA0TE | 00 _H | 8 | FFE80014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA timer start trigger register | ENCA0TS | 00 _H | 8 | FFE80018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA timer stop trigger register | ENCA0TT | 00 _H | 8 | FFE8001C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA I/O control register 0 | ENCA0IOC0 | 00 _H | 8 | FFE80020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA control register | ENCA0CTL | 0000 _H | 16 | FFE80040 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA0 | ENCA I/O control register 1 | ENCA0IOC1 | 00 _H | 8 | FFE80044 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA capture compare register 0 | ENCA1CCR0 | 0000 _H | 16 | FFE81000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA capture compare register 1 | ENCA1CCR1 | 0000 _H | 16 | FFE81004 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA counter register | ENCA1CNT | 0000 _H | 16 | FFE81008 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA status flag register | ENCA1FLG | 00 _H | 8 | FFE8100C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA status flag clear register | ENCA1FGC | 00 _H | 8 | FFE81010 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA timer enable status register | ENCA1TE | 00 _H | 8 | FFE81014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA timer start trigger register | ENCA1TS | 00 _H | 8 | FFE81018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA timer stop trigger register | ENCA1TT | 00 _H | 8 | FFE8101C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA I/O control register 0 | ENCA1IOC0 | 00 _H | 8 | FFE81020 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA control register | ENCA1CTL | 0000 _H | 16 | FFE81040 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ENCA1 | ENCA I/O control register 1 | ENCA1IOC1 | 00 _H | 8 | FFE81044 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA0 | TAPA0 flag register | TAPA0FLG | 0000 _H | 16 | FFE90000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA0 | TAPA0 asynchronous control write enable register | TAPA0ACWE | 00 _H | 8 | FFE90004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA0 | TAPA0 asynchronous control start trigger register | TAPA0ACTS | 00 _H | 8 | FFE90008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA0 | TAPA0 asynchronous control stop trigger register | TAPA0ACTT | 00 _H | 8 | FFE9000C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA0 | TAPA0 Hi-Z start trigger register | TAPA0OPHS | 00 _H | 8 | FFE90014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA0 | TAPA0 Hi-Z stop trigger register | TAPA0OPHT | 00 _H | 8 | FFE90018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA0 | TAPA0 control register 0 | TAPA0CTL0 | 0000 _H | 16 | FFE90020 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA1 | TAPA1 flag register | TAPA1FLG | 0000 _H | 16 | FFE91000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA1 | TAPA1 asynchronous control write enable register | TAPA1ACWE | 00 _H | 8 | FFE91004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA1 | TAPA1 asynchronous control start trigger register | TAPA1ACTS | 00 _H | 8 | FFE91008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (107/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|------------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TAPA1 | TAPA1 asynchronous control stop trigger register | TAPA1ACTT | 00 _H | 8 | FFE9100C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA1 | TAPA1 Hi-Z start trigger register | TAPA1OPHS | 00 _H | 8 | FFE91014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA1 | TAPA1 Hi-Z stop trigger register | TAPA1OPHT | 00 _H | 8 | FFE91018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA1 | TAPA1 control register 0 | TAPA1CTL0 | 0000 _H | 16 | FFE91020 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA2 | TAPA2 flag register | TAPA2FLG | 0000 _H | 16 | FFE92000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA2 | TAPA2 asynchronous control write enable register | TAPA2ACWE | 00 _H | 8 | FFE92004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA2 | TAPA2 asynchronous control start trigger register | TAPA2ACTS | 00 _H | 8 | FFE92008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA2 | TAPA2 asynchronous control stop trigger register | TAPA2ACTT | 00 _H | 8 | FFE9200C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA2 | TAPA2 Hi-Z start trigger register | TAPA2OPHS | 00 _H | 8 | FFE92014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA2 | TAPA2 Hi-Z stop trigger register | TAPA2OPHT | 00 _H | 8 | FFE92018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA2 | TAPA2 control register 0 | TAPA2CTL0 | 0000 _H | 16 | FFE92020 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA3 | TAPA3 flag register | TAPA3FLG | 0000 _H | 16 | FFE93000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA3 | TAPA3 asynchronous control write enable register | TAPA3ACWE | 00 _H | 8 | FFE93004 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA3 | TAPA3 asynchronous control start trigger register | TAPA3ACTS | 00 _H | 8 | FFE93008 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA3 | TAPA3 asynchronous control stop trigger register | TAPA3ACTT | 00 _H | 8 | FFE9300C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA3 | TAPA3 Hi-Z start trigger register | TAPA3OPHS | 00 _H | 8 | FFE93014 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA3 | TAPA3 Hi-Z stop trigger register | TAPA3OPHT | 00 _H | 8 | FFE93018 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TAPA3 | TAPA3 control register 0 | TAPA3CTL0 | 0000 _H | 16 | FFE93020 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF00 | 0000 _H | 16 | FFE00000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF01 | 0000 _H | 16 | FFE00004 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF02 | 0000 _H | 16 | FFE00008 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF03 | 0000 _H | 16 | FFE0000C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF04 | 0000 _H | 16 | FFE00010 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF05 | 0000 _H | 16 | FFE00014 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF06 | 0000 _H | 16 | FFE00018 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF07 | 0000 _H | 16 | FFE0001C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF08 | 0000 _H | 16 | FFE00020 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF09 | 0000 _H | 16 | FFE00024 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF10 | 0000 _H | 16 | FFE00028 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF11 | 0000 _H | 16 | FFE0002C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF12 | 0000 _H | 16 | FFE00030 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF13 | 0000 _H | 16 | FFE00034 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF14 | 0000 _H | 16 | FFE00038 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF15 | 0000 _H | 16 | FFE0003C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF16 | 0000 _H | 16 | FFE00040 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF17 | 0000 _H | 16 | FFE00044 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF18 | 0000 _H | 16 | FFE00048 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF19 | 0000 _H | 16 | FFE0004C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF20 | 0000 _H | 16 | FFE00050 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF21 | 0000 _H | 16 | FFE00054 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF22 | 0000 _H | 16 | FFE00058 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF23 | 0000 _H | 16 | FFE0005C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF24 | 0000 _H | 16 | FFE00060 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF25 | 0000 _H | 16 | FFE00064 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF26 | 0000 _H | 16 | FFE00068 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF27 | 0000 _H | 16 | FFE0006C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF28 | 0000 _H | 16 | FFE00070 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF29 | 0000 _H | 16 | FFE00074 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (108/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---------------------------------------|------------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF30 | 0000 _H | 16 | FFEA0078 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF31 | 0000 _H | 16 | FFEA007C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF32 | 0000 _H | 16 | FFEA0080 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF33 | 0000 _H | 16 | FFEA0084 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF34 | 0000 _H | 16 | FFEA0088 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF35 | 0000 _H | 16 | FFEA008C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF36 | 0000 _H | 16 | FFEA0090 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF37 | 0000 _H | 16 | FFEA0094 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF38 | 0000 _H | 16 | FFEA0098 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF39 | 0000 _H | 16 | FFEA009C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF40 | 0000 _H | 16 | FFEA00A0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF41 | 0000 _H | 16 | FFEA00A4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF42 | 0000 _H | 16 | FFEA00A8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF43 | 0000 _H | 16 | FFEA00AC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF44 | 0000 _H | 16 | FFEA00B0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF45 | 0000 _H | 16 | FFEA00B4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF46 | 0000 _H | 16 | FFEA00B8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF47 | 0000 _H | 16 | FFEA00BC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF48 | 0000 _H | 16 | FFEA00C0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF49 | 0000 _H | 16 | FFEA00C4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF50 | 0000 _H | 16 | FFEA00C8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF51 | 0000 _H | 16 | FFEA00CC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF52 | 0000 _H | 16 | FFEA00D0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF53 | 0000 _H | 16 | FFEA00D4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF54 | 0000 _H | 16 | FFEA00D8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF55 | 0000 _H | 16 | FFEA00DC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF56 | 0000 _H | 16 | FFEA00E0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF57 | 0000 _H | 16 | FFEA00E4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF58 | 0000 _H | 16 | FFEA00E8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF59 | 0000 _H | 16 | FFEA00EC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF60 | 0000 _H | 16 | FFEA00F0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF61 | 0000 _H | 16 | FFEA00F4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF62 | 0000 _H | 16 | FFEA00F8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 duty setting register | TPBA0BUF63 | 0000 _H | 16 | FFEA00FC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 period setting register | TPBA0CMP0 | 0000 _H | 16 | FFEA0100 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 pattern number setting register | TPBA0CMP1 | 00 _H | 8 | FFEA0104 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 timer counter register | TPBA0CNT0 | FFFF _H | 16 | FFEA0108 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 address counter register | TPBA0CNT1 | 00 _H | 8 | FFEA010C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 reload status register | TPBA0RSF | 00 _H | 8 | FFEA0110 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 reload data trigger register | TPBA0RDT | 00 _H | 8 | FFEA0114 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 reload data mode register | TPBA0RDM | 00 _H | 8 | FFEA0118 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 timer output register | TPBA0TO | 00 _H | 8 | FFEA011C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 timer output enable register | TPBA0TOE | 00 _H | 8 | FFEA0120 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 timer output level register | TPBA0TOL | 00 _H | 8 | FFEA0124 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 enable status register | TPBA0TE | 00 _H | 8 | FFEA0128 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 start trigger register | TPBA0TS | 00 _H | 8 | FFEA012C _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 stop trigger register | TPBA0TT | 00 _H | 8 | FFEA0130 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA0 | TPBA0 control register | TPBA0CTL | 00 _H | 8 | FFEA0200 _H | 1 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (109/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------|------------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF00 | 0000 _H | 16 | FFEA1000 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF01 | 0000 _H | 16 | FFEA1004 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF02 | 0000 _H | 16 | FFEA1008 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF03 | 0000 _H | 16 | FFEA100C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF04 | 0000 _H | 16 | FFEA1010 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF05 | 0000 _H | 16 | FFEA1014 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF06 | 0000 _H | 16 | FFEA1018 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF07 | 0000 _H | 16 | FFEA101C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF08 | 0000 _H | 16 | FFEA1020 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF09 | 0000 _H | 16 | FFEA1024 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF10 | 0000 _H | 16 | FFEA1028 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF11 | 0000 _H | 16 | FFEA102C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF12 | 0000 _H | 16 | FFEA1030 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF13 | 0000 _H | 16 | FFEA1034 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF14 | 0000 _H | 16 | FFEA1038 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF15 | 0000 _H | 16 | FFEA103C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF16 | 0000 _H | 16 | FFEA1040 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF17 | 0000 _H | 16 | FFEA1044 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF18 | 0000 _H | 16 | FFEA1048 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF19 | 0000 _H | 16 | FFEA104C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF20 | 0000 _H | 16 | FFEA1050 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF21 | 0000 _H | 16 | FFEA1054 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF22 | 0000 _H | 16 | FFEA1058 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF23 | 0000 _H | 16 | FFEA105C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF24 | 0000 _H | 16 | FFEA1060 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF25 | 0000 _H | 16 | FFEA1064 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF26 | 0000 _H | 16 | FFEA1068 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF27 | 0000 _H | 16 | FFEA106C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF28 | 0000 _H | 16 | FFEA1070 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF29 | 0000 _H | 16 | FFEA1074 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF30 | 0000 _H | 16 | FFEA1078 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF31 | 0000 _H | 16 | FFEA107C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF32 | 0000 _H | 16 | FFEA1080 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF33 | 0000 _H | 16 | FFEA1084 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF34 | 0000 _H | 16 | FFEA1088 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF35 | 0000 _H | 16 | FFEA108C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF36 | 0000 _H | 16 | FFEA1090 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF37 | 0000 _H | 16 | FFEA1094 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF38 | 0000 _H | 16 | FFEA1098 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF39 | 0000 _H | 16 | FFEA109C _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF40 | 0000 _H | 16 | FFEA10A0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF41 | 0000 _H | 16 | FFEA10A4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF42 | 0000 _H | 16 | FFEA10A8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF43 | 0000 _H | 16 | FFEA10AC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF44 | 0000 _H | 16 | FFEA10B0 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF45 | 0000 _H | 16 | FFEA10B4 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF46 | 0000 _H | 16 | FFEA10B8 _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF47 | 0000 _H | 16 | FFEA10BC _H | 1 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (110/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|--|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF48 | 0000 _H | 16 | FFEA10C0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF49 | 0000 _H | 16 | FFEA10C4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF50 | 0000 _H | 16 | FFEA10C8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF51 | 0000 _H | 16 | FFEA10CC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF52 | 0000 _H | 16 | FFEA10D0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF53 | 0000 _H | 16 | FFEA10D4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF54 | 0000 _H | 16 | FFEA10D8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF55 | 0000 _H | 16 | FFEA10DC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF56 | 0000 _H | 16 | FFEA10E0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF57 | 0000 _H | 16 | FFEA10E4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF58 | 0000 _H | 16 | FFEA10E8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF59 | 0000 _H | 16 | FFEA10EC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF60 | 0000 _H | 16 | FFEA10F0 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF61 | 0000 _H | 16 | FFEA10F4 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF62 | 0000 _H | 16 | FFEA10F8 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 duty setting register | TPBA1BUF63 | 0000 _H | 16 | FFEA10FC _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 period setting register | TPBA1CMP0 | 0000 _H | 16 | FFEA1100 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 pattern number setting register | TPBA1CMP1 | 00 _H | 8 | FFEA1104 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 timer counter register | TPBA1CNT0 | FFFF _H | 16 | FFEA1108 _H | 1 | 16 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 address counter register | TPBA1CNT1 | 00 _H | 8 | FFEA110C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 reload status register | TPBA1RSF | 00 _H | 8 | FFEA1110 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 reload data trigger register | TPBA1RDT | 00 _H | 8 | FFEA1114 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 reload data mode register | TPBA1RDM | 00 _H | 8 | FFEA1118 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 timer output register | TPBA1TO | 00 _H | 8 | FFEA111C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 timer output enable register | TPBA1TOE | 00 _H | 8 | FFEA1120 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 timer output level register | TPBA1TOL | 00 _H | 8 | FFEA1124 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 enable status register | TPBA1TE | 00 _H | 8 | FFEA1128 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 start trigger register | TPBA1TS | 00 _H | 8 | FFEA112C _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 stop trigger register | TPBA1TT | 00 _H | 8 | FFEA1130 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| TPBA1 | TPBA1 control register | TPBA1CTL | 00 _H | 8 | FFEA1200 _H | 1 | 8 | √ | √ | √ | √ | √ | √ |
| MODC | Mode Register | MODE | 0000000X _H | 32 | FFF80104 _H | 5 | 32 | — | √ (Terminal Reset only) | — | — | √ | √ |
| SYS | Field BIST control register | BSEQOCTL | 00000001 _H | 32 | FFF80200 _H | 5 | 32 | √ | √ (The registers are initialized in case of CVM reset only.) | — | — | √ | √ |
| SYS | Reset Factor Register | RESF | 00000XXX _H | 32 | FFF81000 _H | 5 | 32 | √ | √ (The register is initialized in case of CVM Reset only. Each bit of RESF is cleared as described in Table 8.6.) | — | — | √ | √ |
| SYS | Reset Factor Clear Register | RESFC | 00000000 _H | 32 | FFF81008 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Software System Reset Request Register 0 | SWSRESA0 | 00000000 _H | 32 | FFF81100 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | Software Application Reset Request Register 0 | SWARES0 | 00000000 _H | 32 | FFF81200 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | RAM Initialization Mode Control Register for DTS-RAM | STAC_DTSRAM | 00000003 _H | 32 | FFF81320 _H | 5 | 32 | √ | √ | √ | — | √ | √ |

Table A.1 List of Registers (111/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|---|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| SYS | RAM Initialization Mode Control Register for Global RAM | STAC_GRAM | 00000003 _H | 32 | FFF81420 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | RAM Initialization Mode Control Register for Local RAM | STAC_LM0 | 00000003 _H | 32 | FFF81520 _H | 5 | 32 | √ | √ (The registers are initialized in case of CVM reset only.) | — | — | √ | √ |
| SYS | RAM Initialization Mode Control Register for CSIH | STAC_LM10 | 00000003 _H | 32 | FFF81E20 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| FLASH | FHVE3 control register | FHVE3 | 00000000 _H | 32 | FFF82410 _H | 5 | 32 | √ | √ | √ | — | √ | √ |
| SYS | RAM Initialization Mode Control Register for CSIH | RESC | 00000001 _H | 32 | FFF82800 _H | 5 | 32 | √ | √ | — | — | √ | √ |
| SYS | CVM Factor Register | CVMF | 00 _H | 8 | FFF82C00 _H | 5 | 8 | √ | — | — | — | √ | √ |
| SYS | CVM Detection Enable Register | CVMDE | 03 _H | 8 | FFF82C04 _H | 5 | 8 | √ | — | — | — | √ | √ |
| SYS | CVM Detection Output Mask Register | CVMDMASK | 00 _H | 8 | FFF82C0C _H | 5 | 8 | √ | √ (CVM Reset is excluded.) | — | — | √ | √ |
| SYS | CVM DIAG Mode Setting Register | CVMDIAG | 00 _H | 8 | FFF82C10 _H | 5 | 8 | √ | √ | — | — | √ | √ |
| SYS | CVM Monitor Register | CVMMON | 0X _H | 8 | FFF82C14 _H | 5 | 8 | — | — | — | — | √ | √ |
| SYS | CVM Factor Clear Register | CVMFC | 00 _H | 8 | FFF82C18 _H | 5 | 8 | — | — | — | — | √ | √ |
| SYS | CVM Detection Enable Set Register | CVMDEW | 00 _H | 8 | FFF82C1C _H | 5 | 8 | — | — | — | — | √ | √ |
| CLMAC | CLMA Self-test Register | CLMATEST | 00000000 _H | 32 | FFF83000 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| CLMAC | CLMA Self-test Status Register | CLMATESTS | 00000000 _H | 32 | FFF83004 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| CLMA0 | CLMA _n Control Register 0 | CLMA0CTL0 | 00 _H | 8 | FFF83100 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA0 | CLMA _n Compare Register L | CLMA0CMPL | 0001 _H | 16 | FFF83108 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA0 | CLMA _n Compare Register H | CLMA0CMPH | 03FF _H | 16 | FFF8310C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA0 | CLMA _n Protection Command Register | CLMA0PCMD | 00 _H | 8 | FFF83110 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA0 | CLMA _n Protection Status Register | CLMA0PS | 00 _H | 8 | FFF83114 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA1 | CLMA _n Control Register 0 | CLMA1CTL0 | 00 _H | 8 | FFF83200 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA1 | CLMA _n Compare Register L | CLMA1CMPL | 0001 _H | 16 | FFF83208 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA1 | CLMA _n Compare Register H | CLMA1CMPH | 03FF _H | 16 | FFF8320C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA1 | CLMA _n Protection Command Register | CLMA1PCMD | 00 _H | 8 | FFF83210 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA1 | CLMA _n Protection Status Register | CLMA1PS | 00 _H | 8 | FFF83214 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA2 | CLMA _n Control Register 0 | CLMA2CTL0 | 00 _H | 8 | FFF83300 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA2 | CLMA _n Compare Register L | CLMA2CMPL | 0001 _H | 16 | FFF83308 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA2 | CLMA _n Compare Register H | CLMA2CMPH | 03FF _H | 16 | FFF8330C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA2 | CLMA _n Protection Command Register | CLMA2PCMD | 00 _H | 8 | FFF83310 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA2 | CLMA _n Protection Status Register | CLMA2PS | 00 _H | 8 | FFF83314 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA3 | CLMA _n Control Register 0 | CLMA3CTL0 | 00 _H | 8 | FFF83400 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA3 | CLMA _n Compare Register L | CLMA3CMPL | 0001 _H | 16 | FFF83408 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA3 | CLMA _n Compare Register H | CLMA3CMPH | 03FF _H | 16 | FFF8340C _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| CLMA3 | CLMA _n Protection Command Register | CLMA3PCMD | 00 _H | 8 | FFF83410 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| CLMA3 | CLMA _n Protection Status Register | CLMA3PS | 00 _H | 8 | FFF83414 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Divider 2 Divisor Register | CLKD2DIV | 00000000 _H | 32 | FFF88810 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Divider 2 Status Register | CLKD2STAT | 00000002 _H | 32 | FFF88814 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Divider 3 Divisor Register | CLKD3DIV | 00000000 _H | 32 | FFF88818 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Divider 3 Status Register | CLKD3STAT | 00000002 _H | 32 | FFF8881C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Selector 2 Control Register | CKSC2C | 00000004 _H | 32 | FFF89080 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Selector 2 Status Register | CKSC2S | 00000004 _H | 32 | FFF89088 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Selector 3 Control Register | CKSC3C | 00000004 _H | 32 | FFF890C0 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Selector 3 Status Register | CKSC3S | 00000004 _H | 32 | FFF890C8 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (112/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| SYS | Clock Selector 8 Control Register | CKSC8C | 00000002 _H | 32 | FFF89110 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Clock Selector 8 Status Register | CKSC8S | 00000003 _H | 32 | FFF89114 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS100C | 00000002 _H | 32 | FFF89120 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC100STAT | 00000003 _H | 32 | FFF89124 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS101C | 00000002 _H | 32 | FFF89128 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC101STAT | 00000003 _H | 32 | FFF8912C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS106C | 00000002 _H | 32 | FFF89130 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC106STAT | 00000003 _H | 32 | FFF89134 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS107C | 00000002 _H | 32 | FFF89138 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC107STAT | 00000003 _H | 32 | FFF8913C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS108C | 00000002 _H | 32 | FFF89140 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC108STAT | 00000003 _H | 32 | FFF89144 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS109C | 00000002 _H | 32 | FFF89148 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC109STAT | 00000003 _H | 32 | FFF8914C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS110C | 00000002 _H | 32 | FFF89150 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC110STAT | 00000003 _H | 32 | FFF89154 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS112C | 00000002 _H | 32 | FFF89158 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC112STAT | 00000003 _H | 32 | FFF8915C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS113C | 00000002 _H | 32 | FFF89160 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC113STAT | 00000003 _H | 32 | FFF89164 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS114C | 00000002 _H | 32 | FFF89168 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC114STAT | 00000003 _H | 32 | FFF8916C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS104C | 00000002 _H | 32 | FFF89170 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC104STAT | 00000003 _H | 32 | FFF89174 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS105C | 00000002 _H | 32 | FFF89178 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC105STAT | 00000003 _H | 32 | FFF8917C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source select register | DNFCKS102C | 00000002 _H | 32 | FFF89180 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| SYS | Digital noise elimination sampling clock source status register | DNFCSC102STAT | 00000003 _H | 32 | FFF89184 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| FLASH | FHVE15 control register | FHVE15 | 00000000 _H | 32 | FFF8A430 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 00 | ADCG0VCR00 | 00000000 _H | 32 | FFF91000 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 01 | ADCG0VCR01 | 00000000 _H | 32 | FFF91004 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 02 | ADCG0VCR02 | 00000000 _H | 32 | FFF91008 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 03 | ADCG0VCR03 | 00000000 _H | 32 | FFF9100C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 04 | ADCG0VCR04 | 00000000 _H | 32 | FFF91010 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 05 | ADCG0VCR05 | 00000000 _H | 32 | FFF91014 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 06 | ADCG0VCR06 | 00000000 _H | 32 | FFF91018 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 07 | ADCG0VCR07 | 00000000 _H | 32 | FFF9101C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 08 | ADCG0VCR08 | 00000000 _H | 32 | FFF91020 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Virtual channel register 09 | ADCG0VCR09 | 00000000 _H | 32 | FFF91024 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (113/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ADCG0 | Virtual channel register 10 | ADCG0VCR10 | 00000000 _H | 32 | FFF91028 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 11 | ADCG0VCR11 | 00000000 _H | 32 | FFF9102C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 12 | ADCG0VCR12 | 00000000 _H | 32 | FFF91030 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 13 | ADCG0VCR13 | 00000000 _H | 32 | FFF91034 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 14 | ADCG0VCR14 | 00000000 _H | 32 | FFF91038 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 15 | ADCG0VCR15 | 00000000 _H | 32 | FFF9103C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 16 | ADCG0VCR16 | 00000000 _H | 32 | FFF91040 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 17 | ADCG0VCR17 | 00000000 _H | 32 | FFF91044 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 18 | ADCG0VCR18 | 00000000 _H | 32 | FFF91048 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 19 | ADCG0VCR19 | 00000000 _H | 32 | FFF9104C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 20 | ADCG0VCR20 | 00000000 _H | 32 | FFF91050 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 21 | ADCG0VCR21 | 00000000 _H | 32 | FFF91054 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 22 | ADCG0VCR22 | 00000000 _H | 32 | FFF91058 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Virtual channel register 23 | ADCG0VCR23 | 00000000 _H | 32 | FFF9105C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 00 | ADCG0DR00 | 00000000 _H | 32 | FFF91100 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 02 | ADCG0DR02 | 00000000 _H | 32 | FFF91104 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 04 | ADCG0DR04 | 00000000 _H | 32 | FFF91108 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 06 | ADCG0DR06 | 00000000 _H | 32 | FFF9110C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 08 | ADCG0DR08 | 00000000 _H | 32 | FFF91110 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 10 | ADCG0DR10 | 00000000 _H | 32 | FFF91114 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 12 | ADCG0DR12 | 00000000 _H | 32 | FFF91118 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 14 | ADCG0DR14 | 00000000 _H | 32 | FFF9111C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 16 | ADCG0DR16 | 00000000 _H | 32 | FFF91120 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 18 | ADCG0DR18 | 00000000 _H | 32 | FFF91124 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 20 | ADCG0DR20 | 00000000 _H | 32 | FFF91128 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data register 22 | ADCG0DR22 | 00000000 _H | 32 | FFF9112C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 00 | ADCG0DIR00 | 00000000 _H | 32 | FFF91200 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 01 | ADCG0DIR01 | 00000000 _H | 32 | FFF91204 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 02 | ADCG0DIR02 | 00000000 _H | 32 | FFF91208 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 03 | ADCG0DIR03 | 00000000 _H | 32 | FFF9120C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 04 | ADCG0DIR04 | 00000000 _H | 32 | FFF91210 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 05 | ADCG0DIR05 | 00000000 _H | 32 | FFF91214 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 06 | ADCG0DIR06 | 00000000 _H | 32 | FFF91218 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 07 | ADCG0DIR07 | 00000000 _H | 32 | FFF9121C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 08 | ADCG0DIR08 | 00000000 _H | 32 | FFF91220 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 09 | ADCG0DIR09 | 00000000 _H | 32 | FFF91224 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 10 | ADCG0DIR10 | 00000000 _H | 32 | FFF91228 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 11 | ADCG0DIR11 | 00000000 _H | 32 | FFF9122C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 12 | ADCG0DIR12 | 00000000 _H | 32 | FFF91230 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 13 | ADCG0DIR13 | 00000000 _H | 32 | FFF91234 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 14 | ADCG0DIR14 | 00000000 _H | 32 | FFF91238 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 15 | ADCG0DIR15 | 00000000 _H | 32 | FFF9123C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 16 | ADCG0DIR16 | 00000000 _H | 32 | FFF91240 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 17 | ADCG0DIR17 | 00000000 _H | 32 | FFF91244 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 18 | ADCG0DIR18 | 00000000 _H | 32 | FFF91248 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 19 | ADCG0DIR19 | 00000000 _H | 32 | FFF9124C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 20 | ADCG0DIR20 | 00000000 _H | 32 | FFF91250 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Data supplementary information register 21 | ADCG0DIR21 | 00000000 _H | 32 | FFF91254 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (114/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-----------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ADCG0 | Data supplementary information register 22 | ADCG0DIR22 | 00000000 _H | 32 | FFF91258 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Data supplementary information register 23 | ADCG0DIR23 | 00000000 _H | 32 | FFF9125C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D synchronization start control register | ADCG0ADSYNSTCR | 00 _H | 8 | FFF91300 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D timer synchronization start control register | ADCG0ADTSYNSTCR | 00 _H | 8 | FFF91304 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D conversion time control register | ADCG0SMPCR | 0000 _H | 16 | FFF91340 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D halt register | ADCG0ADHALTR | 00 _H | 8 | FFF91380 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D control register 1 | ADCG0ADCR1 | 00 _H | 8 | FFF91384 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | MPX current control register | ADCG0MPXCURCR | 00 _H | 8 | FFF91388 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | MPX current register | ADCG0MPXCURR | 00000000 _H | 32 | FFF9138C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | MPX optional wait register | ADCG0MPXOWR | 00 _H | 8 | FFF91390 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D control register 2 | ADCG0ADCR2 | 00 _H | 8 | FFF91398 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D conversion monitor virtual channel pointer 0 | ADCG0ADENDP0 | 00 _H | 8 | FFF913A0 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D conversion monitor virtual channel pointer 1 | ADCG0ADENDP1 | 00 _H | 8 | FFF913A4 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D conversion monitor virtual channel pointer 2 | ADCG0ADENDP2 | 00 _H | 8 | FFF913A8 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D conversion monitor virtual channel pointer 3 | ADCG0ADENDP3 | 00 _H | 8 | FFF913AC _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | A/D conversion monitor virtual channel pointer 4 | ADCG0ADENDP4 | 00 _H | 8 | FFF913B0 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Safety control register | ADCG0SFTCR | 00 _H | 8 | FFF913C0 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Pin level self-diagnostic control register | ADCG0TDCR | 00 _H | 8 | FFF913C4 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Wiring-break detection control register | ADCG0ODCR | 00000000 _H | 32 | FFF913C8 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Upper-limit/lower-limit table register 0 | ADCG0ULLMTBR0 | 7FFE0000 _H | 32 | FFF913CC _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Upper-limit/lower-limit table register 1 | ADCG0ULLMTBR1 | 7FFE0000 _H | 32 | FFF913D0 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Upper-limit/lower-limit table register 2 | ADCG0ULLMTBR2 | 7FFE0000 _H | 32 | FFF913D4 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Error clear register | ADCG0ECR | 00 _H | 8 | FFF913D8 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Upper-limit/lower-limit error register | ADCG0ULER | 00 _H | 8 | FFF913DC _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Overwrite error register | ADCG0OWER | 00 _H | 8 | FFF913E0 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Parity error register | ADCG0PER | 00 _H | 8 | FFF913E4 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | ID error register | ADCG0IDER | 00 _H | 8 | FFF913E8 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H sampling start control register | ADCG0THSMPSTCR | 00 _H | 8 | FFF91400 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H stop control register | ADCG0THSTPCR | 00 _H | 8 | FFF91404 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H control register | ADCG0THCR | 00 _H | 8 | FFF91408 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H group A hold start control register | ADCG0THAHLSTCR | 00 _H | 8 | FFF91410 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H group B hold start control register | ADCG0THBHLSTCR | 00 _H | 8 | FFF91414 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H group A control register | ADCG0THACR | 00 _H | 8 | FFF91420 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H group B control register | ADCG0THBCR | 00 _H | 8 | FFF91424 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H enable register | ADCG0THER | 00 _H | 8 | FFF91430 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | T&H group select register | ADCG0THGSR | 0000 _H | 16 | FFF91434 _H | 5 | 16 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 0 start control register 0 | ADCG0SGSTCR0 | 00 _H | 8 | FFF91480 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 0 control register | ADCG0SGCR0 | 00 _H | 8 | FFF91490 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 0 start virtual channel pointer | ADCG0SGVCSP0 | 00 _H | 8 | FFF91494 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 0 end virtual channel pointer | ADCG0SGVCEP0 | 00 _H | 8 | FFF91498 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 0 multicycle register | ADCG0SGMCCYCR0 | 00 _H | 8 | FFF9149C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 0 status register | ADCG0SGSR0 | 00 _H | 8 | FFF914A4 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 0 upper-limit/lower-limit table select register | ADCG0ULLMSR0 | 00 _H | 8 | FFF914B0 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 1 start control register | ADCG0SGSTCR1 | 00 _H | 8 | FFF91500 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 1 control register | ADCG0SGCR1 | 00 _H | 8 | FFF91510 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 1 start virtual channel pointer | ADCG0SGVCSP1 | 00 _H | 8 | FFF91514 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG0 | Scan group 1 end virtual channel pointer | ADCG0SGVCEP1 | 00 _H | 8 | FFF91518 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (115/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|------------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ADCG0 | Scan group 1 multicycle register | ADCG0SGMCYCR1 | 00 _H | 8 | FFF9151C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 1 status register | ADCG0SGSR1 | 00 _H | 8 | FFF91524 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 1 upper-limit/lower-limit table select register | ADCG0ULLMSR1 | 00 _H | 8 | FFF91530 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 2 start control register | ADCG0SGSTCR2 | 00 _H | 8 | FFF91580 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 2 control register | ADCG0SGCR2 | 00 _H | 8 | FFF91590 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 2 start virtual channel pointer | ADCG0SGVCSP2 | 00 _H | 8 | FFF91594 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 2 end virtual channel pointer | ADCG0SGVCEP2 | 00 _H | 8 | FFF91598 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 2 multicycle register | ADCG0SGMCYCR2 | 00 _H | 8 | FFF9159C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 2 status register | ADCG0SGSR2 | 00 _H | 8 | FFF915A4 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 2 upper-limit/lower-limit table select register | ADCG0ULLMSR2 | 00 _H | 8 | FFF915B0 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 3 start control register | ADCG0SGSTCR3 | 00 _H | 8 | FFF91600 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer 3 start control register | ADCG0ADTSTCR3 | 00 _H | 8 | FFF91608 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer 3 end control register | ADCG0ADTENDCR3 | 00 _H | 8 | FFF9160C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 3 control register | ADCG0SGCR3 | 00 _H | 8 | FFF91610 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 3 start virtual channel pointer | ADCG0SGVCSP3 | 00 _H | 8 | FFF91614 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 3 end virtual channel pointer | ADCG0SGVCEP3 | 00 _H | 8 | FFF91618 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 3 multicycle register | ADCG0SGMCYCR3 | 00 _H | 8 | FFF9161C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 3 status register | ADCG0SGSR3 | 00 _H | 8 | FFF91624 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer initial phase register 3 | ADCG0ADTIPR3 | 00000000 _H | 32 | FFF91628 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer period register 3 | ADCG0ADTPRR3 | 001FFFFFF _H | 32 | FFF9162C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 3 upper-limit/lower-limit table select register | ADCG0ULLMSR3 | 00 _H | 8 | FFF91630 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 4 start control register | ADCG0SGSTCR4 | 00 _H | 8 | FFF91680 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer 4 start control register | ADCG0ADTSTCR4 | 00 _H | 8 | FFF91688 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer 4 end control register | ADCG0ADTENDCR4 | 00 _H | 8 | FFF9168C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 4 control register | ADCG0SGCR4 | 00 _H | 8 | FFF91690 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 4 start virtual channel pointer | ADCG0SGVCSP4 | 00 _H | 8 | FFF91694 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 4 end virtual channel pointer | ADCG0SGVCEP4 | 00 _H | 8 | FFF91698 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 4 multicycle register | ADCG0SGMCYCR4 | 00 _H | 8 | FFF9169C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 4 status register | ADCG0SGSR4 | 00 _H | 8 | FFF916A4 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer initial phase register 4 | ADCG0ADTIPR4 | 00000000 _H | 32 | FFF916A8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | A/D timer period register 4 | ADCG0ADTPRR4 | 001FFFFFF _H | 32 | FFF916AC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Scan group 4 upper-limit/lower-limit table select register | ADCG0ULLMSR4 | 00 _H | 8 | FFF916B0 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG0 | Wiring-break detection pin setting register 0 | ADOPDIG0 | 00000000 _H | 32 | FFF91800 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PIC2B | A/D converter synchronized start trigger register | ADSYNTRG | 00 _H | 8 | FFF91C00 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 00 | ADCG1VCR00 | 00000000 _H | 32 | FFF92000 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 01 | ADCG1VCR01 | 00000000 _H | 32 | FFF92004 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 02 | ADCG1VCR02 | 00000000 _H | 32 | FFF92008 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 03 | ADCG1VCR03 | 00000000 _H | 32 | FFF9200C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 04 | ADCG1VCR04 | 00000000 _H | 32 | FFF92010 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 05 | ADCG1VCR05 | 00000000 _H | 32 | FFF92014 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 06 | ADCG1VCR06 | 00000000 _H | 32 | FFF92018 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 07 | ADCG1VCR07 | 00000000 _H | 32 | FFF9201C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 08 | ADCG1VCR08 | 00000000 _H | 32 | FFF92020 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 09 | ADCG1VCR09 | 00000000 _H | 32 | FFF92024 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 10 | ADCG1VCR10 | 00000000 _H | 32 | FFF92028 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Virtual channel register 11 | ADCG1VCR11 | 00000000 _H | 32 | FFF9202C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (116/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ADCG1 | Virtual channel register 12 | ADCG1VCR12 | 00000000 _H | 32 | FFF92030 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 13 | ADCG1VCR13 | 00000000 _H | 32 | FFF92034 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 14 | ADCG1VCR14 | 00000000 _H | 32 | FFF92038 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 15 | ADCG1VCR15 | 00000000 _H | 32 | FFF9203C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 16 | ADCG1VCR16 | 00000000 _H | 32 | FFF92040 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 17 | ADCG1VCR17 | 00000000 _H | 32 | FFF92044 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 18 | ADCG1VCR18 | 00000000 _H | 32 | FFF92048 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 19 | ADCG1VCR19 | 00000000 _H | 32 | FFF9204C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 20 | ADCG1VCR20 | 00000000 _H | 32 | FFF92050 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 21 | ADCG1VCR21 | 00000000 _H | 32 | FFF92054 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 22 | ADCG1VCR22 | 00000000 _H | 32 | FFF92058 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Virtual channel register 23 | ADCG1VCR23 | 00000000 _H | 32 | FFF9205C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 00 | ADCG1DR00 | 00000000 _H | 32 | FFF92100 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 02 | ADCG1DR02 | 00000000 _H | 32 | FFF92104 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 04 | ADCG1DR04 | 00000000 _H | 32 | FFF92108 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 06 | ADCG1DR06 | 00000000 _H | 32 | FFF9210C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 08 | ADCG1DR08 | 00000000 _H | 32 | FFF92110 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 10 | ADCG1DR10 | 00000000 _H | 32 | FFF92114 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 12 | ADCG1DR12 | 00000000 _H | 32 | FFF92118 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 14 | ADCG1DR14 | 00000000 _H | 32 | FFF9211C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 16 | ADCG1DR16 | 00000000 _H | 32 | FFF92120 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 18 | ADCG1DR18 | 00000000 _H | 32 | FFF92124 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 20 | ADCG1DR20 | 00000000 _H | 32 | FFF92128 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data register 22 | ADCG1DR22 | 00000000 _H | 32 | FFF9212C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 00 | ADCG1DIR00 | 00000000 _H | 32 | FFF92200 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 01 | ADCG1DIR01 | 00000000 _H | 32 | FFF92204 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 02 | ADCG1DIR02 | 00000000 _H | 32 | FFF92208 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 03 | ADCG1DIR03 | 00000000 _H | 32 | FFF9220C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 04 | ADCG1DIR04 | 00000000 _H | 32 | FFF92210 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 05 | ADCG1DIR05 | 00000000 _H | 32 | FFF92214 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 06 | ADCG1DIR06 | 00000000 _H | 32 | FFF92218 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 07 | ADCG1DIR07 | 00000000 _H | 32 | FFF9221C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 08 | ADCG1DIR08 | 00000000 _H | 32 | FFF92220 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 09 | ADCG1DIR09 | 00000000 _H | 32 | FFF92224 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 10 | ADCG1DIR10 | 00000000 _H | 32 | FFF92228 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 11 | ADCG1DIR11 | 00000000 _H | 32 | FFF9222C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 12 | ADCG1DIR12 | 00000000 _H | 32 | FFF92230 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 13 | ADCG1DIR13 | 00000000 _H | 32 | FFF92234 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 14 | ADCG1DIR14 | 00000000 _H | 32 | FFF92238 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 15 | ADCG1DIR15 | 00000000 _H | 32 | FFF9223C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 16 | ADCG1DIR16 | 00000000 _H | 32 | FFF92240 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 17 | ADCG1DIR17 | 00000000 _H | 32 | FFF92244 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 18 | ADCG1DIR18 | 00000000 _H | 32 | FFF92248 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 19 | ADCG1DIR19 | 00000000 _H | 32 | FFF9224C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 20 | ADCG1DIR20 | 00000000 _H | 32 | FFF92250 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 21 | ADCG1DIR21 | 00000000 _H | 32 | FFF92254 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 22 | ADCG1DIR22 | 00000000 _H | 32 | FFF92258 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Data supplementary information register 23 | ADCG1DIR23 | 00000000 _H | 32 | FFF9225C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (117/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ADCG1 | A/D conversion time control register | ADCG1SMPCR | 0000 _H | 16 | FFF92340 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D halt register | ADCG1ADHALTR | 00 _H | 8 | FFF92380 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D control register 1 | ADCG1ADCR1 | 00 _H | 8 | FFF92384 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | MPX current control register | ADCG1MPXCURCR | 00 _H | 8 | FFF92388 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | MPX current register | ADCG1MPXCURR | 00000000 _H | 32 | FFF9238C _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | MPX optional wait register | ADCG1MPXOWR | 00 _H | 8 | FFF92390 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D control register 2 | ADCG1ADCR2 | 00 _H | 8 | FFF92398 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D conversion monitor virtual channel pointer 0 | ADCG1ADENDP0 | 00 _H | 8 | FFF923A0 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D conversion monitor virtual channel pointer 1 | ADCG1ADENDP1 | 00 _H | 8 | FFF923A4 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D conversion monitor virtual channel pointer 2 | ADCG1ADENDP2 | 00 _H | 8 | FFF923A8 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D conversion monitor virtual channel pointer 3 | ADCG1ADENDP3 | 00 _H | 8 | FFF923AC _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | A/D conversion monitor virtual channel pointer 4 | ADCG1ADENDP4 | 00 _H | 8 | FFF923B0 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Safety control register | ADCG1SFTCR | 00 _H | 8 | FFF923C0 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Pin level self-diagnostic control register | ADCG1TDCR | 00 _H | 8 | FFF923C4 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Wiring-break detection control register | ADCG1ODCR | 00000000 _H | 32 | FFF923C8 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Upper-limit/lower-limit table register 0 | ADCG1ULLMTBR0 | 7FFE0000 _H | 32 | FFF923CC _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Upper-limit/lower-limit table register 1 | ADCG1ULLMTBR1 | 7FFE0000 _H | 32 | FFF923D0 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Upper-limit/lower-limit table register 2 | ADCG1ULLMTBR2 | 7FFE0000 _H | 32 | FFF923D4 _H | 5 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Error clear register | ADCG1ECR | 00 _H | 8 | FFF923D8 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Upper-limit/lower-limit error register | ADCG1ULER | 00 _H | 8 | FFF923DC _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Overwrite error register | ADCG1OWER | 00 _H | 8 | FFF923E0 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Parity error register | ADCG1PER | 00 _H | 8 | FFF923E4 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | ID error register | ADCG1IDER | 00 _H | 8 | FFF923E8 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H sampling start control register | ADCG1THSMPSTCR | 00 _H | 8 | FFF92400 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H stop control register | ADCG1THSTPCR | 00 _H | 8 | FFF92404 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H control register | ADCG1THCR | 00 _H | 8 | FFF92408 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H group A hold start control register | ADCG1THAHLSTCR | 00 _H | 8 | FFF92410 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H group B hold start control register | ADCG1THBHLSTCR | 00 _H | 8 | FFF92414 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H group A control register | ADCG1THACR | 00 _H | 8 | FFF92420 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H group B control register | ADCG1THBCR | 00 _H | 8 | FFF92424 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H enable register | ADCG1THER | 00 _H | 8 | FFF92430 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | T&H group select register | ADCG1THGSR | 0000 _H | 16 | FFF92434 _H | 5 | 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 0 start control register 0 | ADCG1SGSTCR0 | 00 _H | 8 | FFF92480 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 0 control register | ADCG1SGCR0 | 00 _H | 8 | FFF92490 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 0 start virtual channel pointer | ADCG1SGVCSP0 | 00 _H | 8 | FFF92494 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 0 end virtual channel pointer | ADCG1SGVCEP0 | 00 _H | 8 | FFF92498 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 0 multicycle register | ADCG1SGMCYCR0 | 00 _H | 8 | FFF9249C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 0 status register | ADCG1SGSR0 | 00 _H | 8 | FFF924A4 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 0 upper-limit/lower-limit table select register | ADCG1ULLMSR0 | 00 _H | 8 | FFF924B0 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 1 start control register | ADCG1SGSTCR1 | 00 _H | 8 | FFF92500 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 1 control register | ADCG1SGCR1 | 00 _H | 8 | FFF92510 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 1 start virtual channel pointer | ADCG1SGVCSP1 | 00 _H | 8 | FFF92514 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 1 end virtual channel pointer | ADCG1SGVCEP1 | 00 _H | 8 | FFF92518 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 1 multicycle register | ADCG1SGMCYCR1 | 00 _H | 8 | FFF9251C _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 1 status register | ADCG1SGSR1 | 00 _H | 8 | FFF92524 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 1 upper-limit/lower-limit table select register | ADCG1ULLMSR1 | 00 _H | 8 | FFF92530 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ADCG1 | Scan group 2 start control register | ADCG1SGSTCR2 | 00 _H | 8 | FFF92580 _H | 5 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (118/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------------|------------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| ADCG1 | Scan group 2 control register | ADCG1SGCR2 | 00 _H | 8 | FFF92590 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 2 start virtual channel pointer | ADCG1SGVCSP2 | 00 _H | 8 | FFF92594 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 2 end virtual channel pointer | ADCG1SGVCEP2 | 00 _H | 8 | FFF92598 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 2 multicycle register | ADCG1SGMCYCR2 | 00 _H | 8 | FFF9259C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 2 status register | ADCG1SGSR2 | 00 _H | 8 | FFF925A4 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 2 upper-limit/lower-limit table select register | ADCG1ULLMSR2 | 00 _H | 8 | FFF925B0 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 3 start control register | ADCG1SGSTCR3 | 00 _H | 8 | FFF92600 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer 3 start control register | ADCG1ADTSTCR3 | 00 _H | 8 | FFF92608 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer 3 end control register | ADCG1ADTENDCR3 | 00 _H | 8 | FFF9260C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 3 control register | ADCG1SGCR3 | 00 _H | 8 | FFF92610 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 3 start virtual channel pointer | ADCG1SGVCSP3 | 00 _H | 8 | FFF92614 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 3 end virtual channel pointer | ADCG1SGVCEP3 | 00 _H | 8 | FFF92618 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 3 multicycle register | ADCG1SGMCYCR3 | 00 _H | 8 | FFF9261C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 3 status register | ADCG1SGSR3 | 00 _H | 8 | FFF92624 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer initial phase register 3 | ADCG1ADTIPR3 | 00000000 _H | 32 | FFF92628 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer period register 3 | ADCG1ADTPRR3 | 001FFFFFF _H | 32 | FFF9262C _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 3 upper-limit/lower-limit table select register | ADCG1ULLMSR3 | 00 _H | 8 | FFF92630 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 4 start control register | ADCG1SGSTCR4 | 00 _H | 8 | FFF92680 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer 4 start control register | ADCG1ADTSTCR4 | 00 _H | 8 | FFF92688 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer 4 end control register | ADCG1ADTENDCR4 | 00 _H | 8 | FFF9268C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 4 control register | ADCG1SGCR4 | 00 _H | 8 | FFF92690 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 4 start virtual channel pointer | ADCG1SGVCSP4 | 00 _H | 8 | FFF92694 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 4 end virtual channel pointer | ADCG1SGVCEP4 | 00 _H | 8 | FFF92698 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 4 multicycle register | ADCG1SGMCYCR4 | 00 _H | 8 | FFF9269C _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 4 status register | ADCG1SGSR4 | 00 _H | 8 | FFF926A4 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer initial phase register 4 | ADCG1ADTIPR4 | 00000000 _H | 32 | FFF926A8 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | A/D timer period register 4 | ADCG1ADTPRR4 | 001FFFFFF _H | 32 | FFF926AC _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Scan group 4 upper-limit/lower-limit table select register | ADCG1ULLMSR4 | 00 _H | 8 | FFF926B0 _H | 5 | 8 | √ | √ | √ | √ | √ | √ |
| ADCG1 | Wiring-break detection pin setting register 1 | ADOPDIG1 | 00000000 _H | 32 | FFF92800 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| TSN0 | Temperature sensor control register | TSN0CR | 00000000 _H | 32 | FFF93000 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| TSN0 | Temperature sensor status register | TSN0STAT | 00000000 _H | 32 | FFF93004 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| TSN0 | Temperature sensor diagnosis control register | TSN0DIAG | 00000000 _H | 32 | FFF93008 _H | 5 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT00 | 0605FE1B _H | 32 | FFF94000 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID00 | FFFFFFFF _H | 32 | FFF94004 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT01 | 0605FE17 _H | 32 | FFF94008 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID01 | FFFFFFFF _H | 32 | FFF9400C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT02 | 0605FE1B _H | 32 | FFF94010 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID02 | FFFFFFFF _H | 32 | FFF94014 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT03 | 0605FE17 _H | 32 | FFF94018 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID03 | FFFFFFFF _H | 32 | FFF9401C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT04 | 0605FE1B _H | 32 | FFF94020 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID04 | FFFFFFFF _H | 32 | FFF94024 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT05 | 0605FE17 _H | 32 | FFF94028 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID05 | FFFFFFFF _H | 32 | FFF9402C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT06 | 0605FE1B _H | 32 | FFF94030 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID06 | FFFFFFFF _H | 32 | FFF94034 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT07 | 0605FE17 _H | 32 | FFF94038 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (119/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID07 | FFFFFFFF _H | 32 | FFF9403C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT08 | 0605FE1B _H | 32 | FFF94040 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID08 | FFFFFFFF _H | 32 | FFF94044 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT09 | 0605FE17 _H | 32 | FFF94048 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID09 | FFFFFFFF _H | 32 | FFF9404C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT10 | 0605FE1B _H | 32 | FFF94050 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID10 | FFFFFFFF _H | 32 | FFF94054 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT11 | 0605FE17 _H | 32 | FFF94058 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID11 | FFFFFFFF _H | 32 | FFF9405C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT12 | 0605FE1B _H | 32 | FFF94060 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID12 | FFFFFFFF _H | 32 | FFF94064 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT13 | 0605FE17 _H | 32 | FFF94068 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID13 | FFFFFFFF _H | 32 | FFF9406C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT14 | 0605FE1B _H | 32 | FFF94070 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID14 | FFFFFFFF _H | 32 | FFF94074 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2APROT15 | 0605FE17 _H | 32 | FFF94078 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2ASPID15 | FFFFFFFF _H | 32 | FFF9407C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT00 | 0605FE1B _H | 32 | FFF94080 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID00 | FFFFFFFF _H | 32 | FFF94084 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT01 | 0605FE17 _H | 32 | FFF94088 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID01 | FFFFFFFF _H | 32 | FFF9408C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT02 | 0605FE1B _H | 32 | FFF94090 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID02 | FFFFFFFF _H | 32 | FFF94094 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT03 | 0605FE17 _H | 32 | FFF94098 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID03 | FFFFFFFF _H | 32 | FFF9409C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT04 | 0605FE1B _H | 32 | FFF940A0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID04 | FFFFFFFF _H | 32 | FFF940A4 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT05 | 0605FE17 _H | 32 | FFF940A8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID05 | FFFFFFFF _H | 32 | FFF940AC _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT06 | 0605FE1B _H | 32 | FFF940B0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID06 | FFFFFFFF _H | 32 | FFF940B4 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT07 | 0605FE17 _H | 32 | FFF940B8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID07 | FFFFFFFF _H | 32 | FFF940BC _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT08 | 0605FE1B _H | 32 | FFF940C0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID08 | FFFFFFFF _H | 32 | FFF940C4 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT09 | 0605FE17 _H | 32 | FFF940C8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID09 | FFFFFFFF _H | 32 | FFF940CC _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT10 | 0605FE1B _H | 32 | FFF940D0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID10 | FFFFFFFF _H | 32 | FFF940D4 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT11 | 0605FE17 _H | 32 | FFF940D8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID11 | FFFFFFFF _H | 32 | FFF940DC _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT12 | 0605FE1B _H | 32 | FFF940E0 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID12 | FFFFFFFF _H | 32 | FFF940E4 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2BPROT13 | 0605FE17 _H | 32 | FFF940E8 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2BSPID13 | FFFFFFFF _H | 32 | FFF940EC _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT00 | 0605FE1B _H | 32 | FFF94100 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID00 | FFFFFFFF _H | 32 | FFF94104 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT01 | 0605FE17 _H | 32 | FFF94108 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (120/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID01 | FFFFFFFF _H | 32 | FFF9410C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT02 | 0605FE1B _H | 32 | FFF94110 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID02 | FFFFFFFF _H | 32 | FFF94114 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT03 | 0605FE17 _H | 32 | FFF94118 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID03 | FFFFFFFF _H | 32 | FFF9411C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT04 | 0605FE1B _H | 32 | FFF94120 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID04 | FFFFFFFF _H | 32 | FFF94124 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT05 | 0605FE17 _H | 32 | FFF94128 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID05 | FFFFFFFF _H | 32 | FFF9412C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT06 | 0605FE1B _H | 32 | FFF94130 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID06 | FFFFFFFF _H | 32 | FFF94134 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT07 | 0605FE17 _H | 32 | FFF94138 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID07 | FFFFFFFF _H | 32 | FFF9413C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT08 | 0605FE1B _H | 32 | FFF94140 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID08 | FFFFFFFF _H | 32 | FFF94144 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT09 | 0605FE17 _H | 32 | FFF94148 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID09 | FFFFFFFF _H | 32 | FFF9414C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT10 | 0605FE1B _H | 32 | FFF94150 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID10 | FFFFFFFF _H | 32 | FFF94154 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-Bus Guard Protection Setting Register | FSGD2CPROT11 | 0605FE17 _H | 32 | FFF94158 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | P-bus Guard SPID Setting Register | FSGD2CSPID11 | FFFFFFFF _H | 32 | FFF9415C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | ERRSLV Control Register for Pbus Guard | ERRSLV2CTL | 00 _H | 8 | FFF94400 _H | 2 | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | ERRSLV Status Register for PBus Guard | ERRSLV2STAT | 00000000 _H | 32 | FFF94404 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBG2 | ERRSLV Error Transfer Type Register for P-Bus Guard | ERRSLV2TYPE | 00000000 _H | 32 | FFF9440C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC2 | P-Bus Data and Address ECC Control Register | APEC1ECCCTL | 00000000 _H | 32 | FFF94800 _H | 2 | 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC2 | P-Bus Error Information Control Register | APEC1ERRINT | 00000073 _H | 32 | FFF94804 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC2 | P-Bus Data and Address ECC SED/DED Status Clear Register | APEC1STCLR | 00000000 _H | 32 | FFF94808 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC2 | P-Bus Data and Address Error Count Overflow Status Register | APEC1OVFSTR | 00000000 _H | 32 | FFF9480C _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC2 | P-Bus Data and Address ECC SED/DED Status Register | APEC11STERSTR | 00000000 _H | 32 | FFF94810 _H | 2 | 8, 16, 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PBECC2 | P-Bus Data and Address ECC SED/DED Address Register | APEC11STEADR0 | 00000000 _H | 32 | FFF94850 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 0 | PINT0 | 00000000 _H | 32 | FFF98000 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 1 | PINT1 | 00000000 _H | 32 | FFF98004 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 2 | PINT2 | 00000000 _H | 32 | FFF98008 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 3 | PINT3 | 00000000 _H | 32 | FFF9800C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 4 | PINT4 | 00000000 _H | 32 | FFF98010 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 5 | PINT5 | 00000000 _H | 32 | FFF98014 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 6 | PINT6 | 00000000 _H | 32 | FFF98018 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Status Register 7 | PINT7 | 00000000 _H | 32 | FFF9801C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 0 | PINTCLR0 | 00000000 _H | 32 | FFF98020 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 1 | PINTCLR1 | 00000000 _H | 32 | FFF98024 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 2 | PINTCLR2 | 00000000 _H | 32 | FFF98028 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 3 | PINTCLR3 | 00000000 _H | 32 | FFF9802C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 4 | PINTCLR4 | 00000000 _H | 32 | FFF98030 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 5 | PINTCLR5 | 00000000 _H | 32 | FFF98034 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 6 | PINTCLR6 | 00000000 _H | 32 | FFF98038 _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DTSMRG | DTS Interrupt Clear Register 7 | PINTCLR7 | 00000000 _H | 32 | FFF9803C _H | 2 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (121/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMATRG | DMAC Primary/Secondary Select Register 0 | DMACTRGSEL0 | 00000000 _H | 32 | FFF99000 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| DMATRG | DMAC Primary/Secondary Select Register 1 | DMACTRGSEL1 | 00000000 _H | 32 | FFF99004 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| DMATRG | DTS Primary/Secondary Select Register 0 | DTSTRGSEL0 | 00000000 _H | 32 | FFF99008 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| DMATRG | DTS Primary/Secondary Select Register 1 | DTSTRGSEL1 | 00000000 _H | 32 | FFF9900C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus Guard for Read | FSGDF0PROT00 | 0605FE1B _H | 32 | FFFA0000 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus Guard for Read | FSGDF0SPID00 | FFFFFFF _H | 32 | FFFA0004 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus Guard for Write | FSGDF0PROT01 | 0605FE17 _H | 32 | FFFA0008 _H | 2 | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus Guard for Write | FSGDF0SPID01 | FFFFFFF _H | 32 | FFFA000C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus Guard Error slave | ERRSLVF0CTL | 00 _H | 8 | FFFA0010 _H | 2 | 8 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus Guard Error slave | ERRSLVF0STAT | 00000000 _H | 32 | FFFA0014 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus Guard Error slave | ERRSLVF0TYPE | 00000000 _H | 32 | FFFA001C _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| HBG | Flexray0 H-bus master SPID setting | HSSPIDRG0 | 00000000 _H | 32 | FFFA1000 _H | 2 | 32 | √ | √ | √ | √ | √ | √ |
| IPG | Peripherals Protection Violation Access Information Register | IPGECRUM | Undefined (retained) | 16 | FFFE002 _H | CPU | 16 | √ | √ | √ | √ | √ | — |
| IPG | Peripherals Protection Violation Access Address Register | IPGADRUM | Undefined (retained) | 32 | FFFE008 _H | CPU | 32 | √ | √ | √ | √ | √ | — |
| IPG | Peripherals Protection Enable Register | IPGENUM | 00 _H | 8 | FFFE00D _H | CPU | 8 | √ | √ | √ | √ | √ | — |
| IPG | Peripherals Protection Setting Register 0 | IPGPMTUM0 | 00 _H | 8 | FFFE020 _H | CPU | 8 | √ | √ | √ | √ | √ | — |
| IPG | Peripherals Protection Setting Register 1 | IPGPMTUM1 | 00 _H | 8 | FFFE021 _H | CPU | 8 | √ | √ | √ | √ | √ | — |
| IPG | Peripherals Protection Setting Register 2 | IPGPMTUM2 | 00 _H | 8 | FFFE022 _H | CPU | 8 | √ | √ | √ | √ | √ | — |
| IPG | Peripherals Protection Setting Register 3 | IPGPMTUM3 | 00 _H | 8 | FFFE023 _H | CPU | 8 | √ | √ | √ | √ | √ | — |
| IPG | Peripherals Protection Setting Register 4 | IPGPMTUM4 | 00 _H | 8 | FFFE024 _H | CPU | 8 | √ | √ | √ | √ | √ | — |
| PEG | PE Guard Area 0 Mask Setting Register | PEGG0MK | 003FF000 _H | 32 | FFFE680 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 0 Base Setting Register | PEGG0BA | FE800003 _H | 32 | FFFE684 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 0 enable Setting Register | PEGG0SP | 00000003 _H | 32 | FFFE688 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 1 Mask Setting Register | PEGG1MK | 00000000 _H | 32 | FFFE690 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 1 Base Setting Register | PEGG1BA | 00000000 _H | 32 | FFFE694 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 1 enable Setting Register | PEGG1SP | 00000000 _H | 32 | FFFE698 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 2 Mask Setting Register | PEGG2MK | 00000000 _H | 32 | FFFE6A0 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 2 Base Setting Register | PEGG2BA | 00000000 _H | 32 | FFFE6A4 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 2 enable Setting Register | PEGG2SP | 00000000 _H | 32 | FFFE6A8 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 3 Mask Setting Register | PEGG3MK | 00000000 _H | 32 | FFFE6B0 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 3 Base Setting Register | PEGG3BA | 00000000 _H | 32 | FFFE6B4 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 3 enable Setting Register | PEGG3SP | 00000000 _H | 32 | FFFE6B8 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 4 Mask Setting Register | PEGG4MK | 00000000 _H | 32 | FFFE6C0 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 4 Base Setting Register | PEGG4BA | 00000000 _H | 32 | FFFE6C4 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 4 enable Setting Register | PEGG4SP | 00000000 _H | 32 | FFFE6C8 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 5 Mask Setting Register | PEGG5MK | 00000000 _H | 32 | FFFE6D0 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 5 Base Setting Register | PEGG5BA | 00000000 _H | 32 | FFFE6D4 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 5 enable Setting Register | PEGG5SP | 00000000 _H | 32 | FFFE6D8 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 6 Mask Setting Register | PEGG6MK | 00000000 _H | 32 | FFFE6E0 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 6 Base Setting Register | PEGG6BA | 00000000 _H | 32 | FFFE6E4 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 6 enable Setting Register | PEGG6SP | 00000000 _H | 32 | FFFE6E8 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 7 Mask Setting Register | PEGG7MK | 00000000 _H | 32 | FFFE6F0 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 7 Base Setting Register | PEGG7BA | 00000000 _H | 32 | FFFE6F4 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| PEG | PE Guard Area 7 enable Setting Register | PEGG7SP | 00000000 _H | 32 | FFFE6F8 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| SEG | Error Notification Control Register | SEGCONT | 0000 _H | 16 | FFFE980 _H | CPU | 16 | √ | √ | √ | √ | √ | R: √ W: — |
| SEG | Error Occurrence Retention Register | SEGFLAG | 0000 _H | 16 | FFFE982 _H | CPU | 16 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (122/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|---------|----------------------|----------------|------------|---------------------|--------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| SEG | SEG Error Information Register (Address) | SEGADDR | Undefined (retained) | 32 | FFFE988 H | CPU | 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC8 | 008F H | 16 | FFFEA10 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC9 | 008F H | 16 | FFFEA12 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC16 | 008F H | 16 | FFFEA20 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC17 | 008F H | 16 | FFFEA22 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC18 | 008F H | 16 | FFFEA24 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC19 | 008F H | 16 | FFFEA26 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC20 | 008F H | 16 | FFFEA28 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC21 | 008F H | 16 | FFFEA2A H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC22 | 008F H | 16 | FFFEA2C H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC23 | 008F H | 16 | FFFEA2E H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC24 | 008F H | 16 | FFFEA30 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC25 | 008F H | 16 | FFFEA32 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC26 | 008F H | 16 | FFFEA34 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC27 | 008F H | 16 | FFFEA36 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC28 | 008F H | 16 | FFFEA38 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC29 | 008F H | 16 | FFFEA3A H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC30 | 008F H | 16 | FFFEA3C H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Control Registers | EIC31 | 008F H | 16 | FFFEA3E H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | FE Level NMI Status Register | FNC | 0000 H | 16 | FFFEA78 H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | FE Level Maskable Interrupt Status Register | FIC | 0000 H | 16 | FFFEA7A H | CPU | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Mask Registers | IMR0 | FFFFFFF H | 32 | FFFEAF0 H | CPU | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD8 | 00000001 H | 32 | FFFEEB20 H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD9 | 00000001 H | 32 | FFFEEB24 H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD16 | 00000001 H | 32 | FFFEEB40 H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD17 | 00000001 H | 32 | FFFEEB44 H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD18 | 00000001 H | 32 | FFFEEB48 H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (123/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|----------|---|--------------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC1 | EI Level Interrupt Bind Registers | EIBD19 | 00000001 _H | 32 | FFFEEB4C _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD20 | 00000001 _H | 32 | FFFEEB50 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD21 | 00000001 _H | 32 | FFFEEB54 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD22 | 00000001 _H | 32 | FFFEEB58 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD23 | 00000001 _H | 32 | FFFEEB5C _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD24 | 00000001 _H | 32 | FFFEEB60 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD25 | 00000001 _H | 32 | FFFEEB64 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD26 | 00000001 _H | 32 | FFFEEB68 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD27 | 00000001 _H | 32 | FFFEEB6C _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD28 | 00000001 _H | 32 | FFFEEB70 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD29 | 00000001 _H | 32 | FFFEEB74 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD30 | 00000001 _H | 32 | FFFEEB78 _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC1 | EI Level Interrupt Bind Registers | EIBD31 | 00000001 _H | 32 | FFFEEB7C _H | CPU | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| TESTCOMP | Comparator Test Register 0 | CMPTST0 | 00000000 _H | 32 | FFFEEED0 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| TESTCOMP | Comparator Test Register 1 | CMPTST1 | 00000000 _H | 32 | FFFEEED4 _H | CPU | 8, 16, 32 | √ | √ | √ | √ | √ | √ |
| CFU | Cache Clear Operation Register | TM_CC | 00000000 _H | 32 | FFFF7808 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| CFU | Tuning Memory Mapping Enable Register | TM_ME | 00000000 _H | 32 | FFFF7810 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| CFU | Tuning Memory Mapping Status Register | TM_MS | 00000000 _H | 32 | FFFF7814 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| CFU | Tuning Memory Mapping Address Registers 0 | TM_MA0 | 00000000 _H | 32 | FFFF7840 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| CFU | Tuning Memory Mapping Address Registers 1 | TM_MA1 | 00000000 _H | 32 | FFFF7844 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| CFU | Tuning Memory Mapping Address Registers 2 | TM_MA2 | 00000000 _H | 32 | FFFF7848 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| CFU | Tuning Memory Mapping Address Registers 3 | TM_MA3 | 00000000 _H | 32 | FFFF784C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRAMC | GRAMC WTBuf Configuration Register 0 | GRAMCWTCFIG0 | 00000001 _H | 32 | FFFF7A04 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRAMC | GRAMC WTBuf Configuration Register 1 | GRAMCWTCFIG1 | 00000007 _H | 32 | FFFF7A08 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| GRAMC | GRAMC WTBuf Configuration Register 2 | GRAMCWTCFIG2 | 00000000 _H | 32 | FFFF7A0C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMA Control Register | DMACTL | 00000000 _H | 32 | FFFF8000 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Control Register 1 | DTSTL1 | 00000000 _H | 32 | FFFF8010 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Control Register 2 | DTSTL2 | 00000000 _H | 32 | FFFF8014 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Status Register | DTSSTS | 00000000 _H | 32 | FFFF8018 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Error Register | DMACER | 00000000 _H | 32 | FFFF8020 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Error Register 1 | DTSER1 | 00000000 _H | 32 | FFFF8024 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Error Register 2 | DTSER2 | 00000000 _H | 32 | FFFF8028 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Error Clear Register | DTSERC | 00000000 _H | 32 | FFFF802C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC0 Register Access Protection Violation Register | DM0CMV | 00000000 _H | 32 | FFFF8030 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC1 Register Access Protection Violation Register | DM1CMV | 00000000 _H | 32 | FFFF8034 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Register Access Protection Violation Register | DTSCMV | 00000000 _H | 32 | FFFF8038 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | Register Access Protection Violation Clear Register | CMVC | 00000000 _H | 32 | FFFF803C _H | 0 | 32 | √ | √ | √ | √ | √ | — |

Table A.1 List of Registers (124/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | Transfer Status Register | TFRSTS | 00000000 _H | 32 | FFFF804C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR0 | 00000000 _H | 32 | FFFF8060 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR1 | 00000000 _H | 32 | FFFF8064 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR2 | 00000000 _H | 32 | FFFF8068 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR3 | 00000000 _H | 32 | FFFF806C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR4 | 00000000 _H | 32 | FFFF8070 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR5 | 00000000 _H | 32 | FFFF8074 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR6 | 00000000 _H | 32 | FFFF8078 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Priority Setting | DTSPR7 | 00000000 _H | 32 | FFFF807C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTSRAM ECC Control Register | DTRRECTL | 00000000 _H | 32 | FFFF8080 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTSRAM Error Notification Control Register | DTRERINT | 00000002 _H | 32 | FFFF8084 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTSRAM Test Control Register | DTRTCTCL | 00000000 _H | 32 | FFFF8094 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTSRAM Test Write Data Register | DTRTWDAT | 00000000 _H | 32 | FFFF8098 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTSRAM Test Read Data Register | DTRTRDAT | 00000000 _H | 32 | FFFF809C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | ECC on BUS Address ECC Test Control Register | ADECCCTL | 00000000 _H | 32 | FFFF80A0 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | ECC on BUS Address ECC Test Data Register | ADECCTDT | 00000000 _H | 32 | FFFF80A4 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM00CM | 00002008 _H | 32 | FFFF8100 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM01CM | 00002008 _H | 32 | FFFF8104 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM02CM | 00002008 _H | 32 | FFFF8108 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM03CM | 00002008 _H | 32 | FFFF810C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM04CM | 00002008 _H | 32 | FFFF8110 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM05CM | 00002008 _H | 32 | FFFF8114 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM06CM | 00002008 _H | 32 | FFFF8118 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM07CM | 00002008 _H | 32 | FFFF811C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM10CM | 00002008 _H | 32 | FFFF8120 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM11CM | 00002008 _H | 32 | FFFF8124 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM12CM | 00002008 _H | 32 | FFFF8128 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM13CM | 00002008 _H | 32 | FFFF812C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM14CM | 00002008 _H | 32 | FFFF8130 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM15CM | 00002008 _H | 32 | FFFF8134 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM16CM | 00002008 _H | 32 | FFFF8138 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DMAC Channel Master Setting | DM17CM | 00002008 _H | 32 | FFFF813C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS000CM | Undefined | 32 | FFFF8200 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS001CM | Undefined | 32 | FFFF8204 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS002CM | Undefined | 32 | FFFF8208 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS003CM | Undefined | 32 | FFFF820C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS004CM | Undefined | 32 | FFFF8210 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS005CM | Undefined | 32 | FFFF8214 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS006CM | Undefined | 32 | FFFF8218 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS007CM | Undefined | 32 | FFFF821C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS008CM | Undefined | 32 | FFFF8220 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS009CM | Undefined | 32 | FFFF8224 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS010CM | Undefined | 32 | FFFF8228 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS011CM | Undefined | 32 | FFFF822C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS012CM | Undefined | 32 | FFFF8230 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS013CM | Undefined | 32 | FFFF8234 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS014CM | Undefined | 32 | FFFF8238 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS015CM | Undefined | 32 | FFFF823C _H | 0 | 32 | √ | √ | √ | √ | √ | — |

Table A.1 List of Registers (125/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|-------------------------------------|----------|-------------------|----------------|----------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTS Channel Master Setting Register | DTS016CM | Undefined | 32 | FFFF8240 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS017CM | Undefined | 32 | FFFF8244 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS018CM | Undefined | 32 | FFFF8248 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS019CM | Undefined | 32 | FFFF824C | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS020CM | Undefined | 32 | FFFF8250 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS021CM | Undefined | 32 | FFFF8254 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS022CM | Undefined | 32 | FFFF8258 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS023CM | Undefined | 32 | FFFF825C | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS024CM | Undefined | 32 | FFFF8260 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS025CM | Undefined | 32 | FFFF8264 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS026CM | Undefined | 32 | FFFF8268 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS027CM | Undefined | 32 | FFFF826C | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS028CM | Undefined | 32 | FFFF8270 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS029CM | Undefined | 32 | FFFF8274 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS030CM | Undefined | 32 | FFFF8278 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS031CM | Undefined | 32 | FFFF827C | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS032CM | Undefined | 32 | FFFF8280 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS033CM | Undefined | 32 | FFFF8284 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS034CM | Undefined | 32 | FFFF8288 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS035CM | Undefined | 32 | FFFF828C | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS036CM | Undefined | 32 | FFFF8290 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS037CM | Undefined | 32 | FFFF8294 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS038CM | Undefined | 32 | FFFF8298 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS039CM | Undefined | 32 | FFFF829C | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS040CM | Undefined | 32 | FFFF82A0 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS041CM | Undefined | 32 | FFFF82A4 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS042CM | Undefined | 32 | FFFF82A8 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS043CM | Undefined | 32 | FFFF82AC | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS044CM | Undefined | 32 | FFFF82B0 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS045CM | Undefined | 32 | FFFF82B4 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS046CM | Undefined | 32 | FFFF82B8 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS047CM | Undefined | 32 | FFFF82BC | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS048CM | Undefined | 32 | FFFF82C0 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS049CM | Undefined | 32 | FFFF82C4 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS050CM | Undefined | 32 | FFFF82C8 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS051CM | Undefined | 32 | FFFF82CC | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS052CM | Undefined | 32 | FFFF82D0 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS053CM | Undefined | 32 | FFFF82D4 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS054CM | Undefined | 32 | FFFF82D8 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS055CM | Undefined | 32 | FFFF82DC | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS056CM | Undefined | 32 | FFFF82E0 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS057CM | Undefined | 32 | FFFF82E4 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS058CM | Undefined | 32 | FFFF82E8 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS059CM | Undefined | 32 | FFFF82EC | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS060CM | Undefined | 32 | FFFF82F0 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS061CM | Undefined | 32 | FFFF82F4 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS062CM | Undefined | 32 | FFFF82F8 | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS063CM | Undefined | 32 | FFFF82FC | H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |

Table A.1 List of Registers (126/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-------------------------------------|----------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTS Channel Master Setting Register | DTS064CM | Undefined | 32 | FFFF8300 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS065CM | Undefined | 32 | FFFF8304 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS066CM | Undefined | 32 | FFFF8308 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS067CM | Undefined | 32 | FFFF830C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS068CM | Undefined | 32 | FFFF8310 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS069CM | Undefined | 32 | FFFF8314 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS070CM | Undefined | 32 | FFFF8318 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS071CM | Undefined | 32 | FFFF831C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS072CM | Undefined | 32 | FFFF8320 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS073CM | Undefined | 32 | FFFF8324 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS074CM | Undefined | 32 | FFFF8328 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS075CM | Undefined | 32 | FFFF832C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS076CM | Undefined | 32 | FFFF8330 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS077CM | Undefined | 32 | FFFF8334 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS078CM | Undefined | 32 | FFFF8338 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS079CM | Undefined | 32 | FFFF833C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS080CM | Undefined | 32 | FFFF8340 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS081CM | Undefined | 32 | FFFF8344 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS082CM | Undefined | 32 | FFFF8348 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS083CM | Undefined | 32 | FFFF834C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS084CM | Undefined | 32 | FFFF8350 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS085CM | Undefined | 32 | FFFF8354 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS086CM | Undefined | 32 | FFFF8358 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS087CM | Undefined | 32 | FFFF835C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS088CM | Undefined | 32 | FFFF8360 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS089CM | Undefined | 32 | FFFF8364 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS090CM | Undefined | 32 | FFFF8368 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS091CM | Undefined | 32 | FFFF836C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS092CM | Undefined | 32 | FFFF8370 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS093CM | Undefined | 32 | FFFF8374 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS094CM | Undefined | 32 | FFFF8378 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS095CM | Undefined | 32 | FFFF837C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS096CM | Undefined | 32 | FFFF8380 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS097CM | Undefined | 32 | FFFF8384 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS098CM | Undefined | 32 | FFFF8388 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS099CM | Undefined | 32 | FFFF838C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS100CM | Undefined | 32 | FFFF8390 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS101CM | Undefined | 32 | FFFF8394 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS102CM | Undefined | 32 | FFFF8398 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS103CM | Undefined | 32 | FFFF839C _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS104CM | Undefined | 32 | FFFF83A0 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS105CM | Undefined | 32 | FFFF83A4 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS106CM | Undefined | 32 | FFFF83A8 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS107CM | Undefined | 32 | FFFF83AC _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS108CM | Undefined | 32 | FFFF83B0 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS109CM | Undefined | 32 | FFFF83B4 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS110CM | Undefined | 32 | FFFF83B8 _H | 0 | 32 | √ | √ | √ | √ | √ | — |
| DMASS | DTS Channel Master Setting Register | DTS111CM | Undefined | 32 | FFFF83BC _H | 0 | 32 | √ | √ | √ | √ | √ | — |

Table A.1 List of Registers (127/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTS Channel Master Setting Register | DTS112CM | Undefined | 32 | FFFF83C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS113CM | Undefined | 32 | FFFF83C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS114CM | Undefined | 32 | FFFF83C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS115CM | Undefined | 32 | FFFF83CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS116CM | Undefined | 32 | FFFF83D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS117CM | Undefined | 32 | FFFF83D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS118CM | Undefined | 32 | FFFF83D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS119CM | Undefined | 32 | FFFF83DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS120CM | Undefined | 32 | FFFF83E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS121CM | Undefined | 32 | FFFF83E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS122CM | Undefined | 32 | FFFF83E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS123CM | Undefined | 32 | FFFF83EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DTS Channel Master Setting Register | DTS124CM | Undefined | 32 | FFFF83F0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | — |
| DMASS | DMAC Source Address Register | DSA0 | 00000000 _H | 32 | FFFF8400 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA0 | 00000000 _H | 32 | FFFF8404 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC0 | 00000000 _H | 32 | FFFF8408 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT0 | 00000000 _H | 32 | FFFF840C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA0 | 00000000 _H | 32 | FFFF8410 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA0 | 00000000 _H | 32 | FFFF8414 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC0 | 00000000 _H | 32 | FFFF8418 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC0 | 00000000 _H | 32 | FFFF841C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN0 | 00000000 _H | 32 | FFFF8420 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST0 | 00000000 _H | 32 | FFFF8424 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS0 | 00000000 _H | 32 | FFFF8428 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC0 | 00000000 _H | 32 | FFFF842C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR0 | 00000000 _H | 32 | FFFF8430 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ0 | 00000000 _H | 32 | FFFF8434 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRC0 | 00000000 _H | 32 | FFFF8438 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA1 | 00000000 _H | 32 | FFFF8440 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA1 | 00000000 _H | 32 | FFFF8444 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC1 | 00000000 _H | 32 | FFFF8448 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT1 | 00000000 _H | 32 | FFFF844C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA1 | 00000000 _H | 32 | FFFF8450 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA1 | 00000000 _H | 32 | FFFF8454 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC1 | 00000000 _H | 32 | FFFF8458 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC1 | 00000000 _H | 32 | FFFF845C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN1 | 00000000 _H | 32 | FFFF8460 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST1 | 00000000 _H | 32 | FFFF8464 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS1 | 00000000 _H | 32 | FFFF8468 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC1 | 00000000 _H | 32 | FFFF846C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR1 | 00000000 _H | 32 | FFFF8470 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ1 | 00000000 _H | 32 | FFFF8474 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRC1 | 00000000 _H | 32 | FFFF8478 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA2 | 00000000 _H | 32 | FFFF8480 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA2 | 00000000 _H | 32 | FFFF8484 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC2 | 00000000 _H | 32 | FFFF8488 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT2 | 00000000 _H | 32 | FFFF848C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA2 | 00000000 _H | 32 | FFFF8490 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (128/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|---------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DMAC Reload Destination Address Register | DRDA2 | 00000000 _H | 32 | FFFF8494 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC2 | 00000000 _H | 32 | FFFF8498 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC2 | 00000000 _H | 32 | FFFF849C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN2 | 00000000 _H | 32 | FFFF84A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST2 | 00000000 _H | 32 | FFFF84A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS2 | 00000000 _H | 32 | FFFF84A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC2 | 00000000 _H | 32 | FFFF84AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR2 | 00000000 _H | 32 | FFFF84B0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ2 | 00000000 _H | 32 | FFFF84B4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRQC2 | 00000000 _H | 32 | FFFF84B8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA3 | 00000000 _H | 32 | FFFF84C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA3 | 00000000 _H | 32 | FFFF84C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC3 | 00000000 _H | 32 | FFFF84C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT3 | 00000000 _H | 32 | FFFF84CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA3 | 00000000 _H | 32 | FFFF84D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA3 | 00000000 _H | 32 | FFFF84D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC3 | 00000000 _H | 32 | FFFF84D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC3 | 00000000 _H | 32 | FFFF84DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN3 | 00000000 _H | 32 | FFFF84E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST3 | 00000000 _H | 32 | FFFF84E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS3 | 00000000 _H | 32 | FFFF84E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC3 | 00000000 _H | 32 | FFFF84EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR3 | 00000000 _H | 32 | FFFF84F0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ3 | 00000000 _H | 32 | FFFF84F4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRQC3 | 00000000 _H | 32 | FFFF84F8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA4 | 00000000 _H | 32 | FFFF8500 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA4 | 00000000 _H | 32 | FFFF8504 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC4 | 00000000 _H | 32 | FFFF8508 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT4 | 00000000 _H | 32 | FFFF850C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA4 | 00000000 _H | 32 | FFFF8510 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA4 | 00000000 _H | 32 | FFFF8514 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC4 | 00000000 _H | 32 | FFFF8518 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC4 | 00000000 _H | 32 | FFFF851C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN4 | 00000000 _H | 32 | FFFF8520 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST4 | 00000000 _H | 32 | FFFF8524 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS4 | 00000000 _H | 32 | FFFF8528 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC4 | 00000000 _H | 32 | FFFF852C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR4 | 00000000 _H | 32 | FFFF8530 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ4 | 00000000 _H | 32 | FFFF8534 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRQC4 | 00000000 _H | 32 | FFFF8538 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA5 | 00000000 _H | 32 | FFFF8540 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA5 | 00000000 _H | 32 | FFFF8544 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC5 | 00000000 _H | 32 | FFFF8548 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT5 | 00000000 _H | 32 | FFFF854C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA5 | 00000000 _H | 32 | FFFF8550 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA5 | 00000000 _H | 32 | FFFF8554 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC5 | 00000000 _H | 32 | FFFF8558 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC5 | 00000000 _H | 32 | FFFF855C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (129/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN5 | 00000000 _H | 32 | FFFF8560 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST5 | 00000000 _H | 32 | FFFF8564 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS5 | 00000000 _H | 32 | FFFF8568 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC5 | 00000000 _H | 32 | FFFF856C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR5 | 00000000 _H | 32 | FFFF8570 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ5 | 00000000 _H | 32 | FFFF8574 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRQC5 | 00000000 _H | 32 | FFFF8578 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA6 | 00000000 _H | 32 | FFFF8580 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA6 | 00000000 _H | 32 | FFFF8584 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC6 | 00000000 _H | 32 | FFFF8588 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT6 | 00000000 _H | 32 | FFFF858C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA6 | 00000000 _H | 32 | FFFF8590 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA6 | 00000000 _H | 32 | FFFF8594 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC6 | 00000000 _H | 32 | FFFF8598 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC6 | 00000000 _H | 32 | FFFF859C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN6 | 00000000 _H | 32 | FFFF85A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST6 | 00000000 _H | 32 | FFFF85A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS6 | 00000000 _H | 32 | FFFF85A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC6 | 00000000 _H | 32 | FFFF85AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR6 | 00000000 _H | 32 | FFFF85B0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ6 | 00000000 _H | 32 | FFFF85B4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRQC6 | 00000000 _H | 32 | FFFF85B8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA7 | 00000000 _H | 32 | FFFF85C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA7 | 00000000 _H | 32 | FFFF85C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC7 | 00000000 _H | 32 | FFFF85C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT7 | 00000000 _H | 32 | FFFF85CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA7 | 00000000 _H | 32 | FFFF85D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA7 | 00000000 _H | 32 | FFFF85D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC7 | 00000000 _H | 32 | FFFF85D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC7 | 00000000 _H | 32 | FFFF85DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN7 | 00000000 _H | 32 | FFFF85E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST7 | 00000000 _H | 32 | FFFF85E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS7 | 00000000 _H | 32 | FFFF85E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC7 | 00000000 _H | 32 | FFFF85EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR7 | 00000000 _H | 32 | FFFF85F0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ7 | 00000000 _H | 32 | FFFF85F4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRQC7 | 00000000 _H | 32 | FFFF85F8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA8 | 00000000 _H | 32 | FFFF8600 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA8 | 00000000 _H | 32 | FFFF8604 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC8 | 00000000 _H | 32 | FFFF8608 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT8 | 00000000 _H | 32 | FFFF860C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA8 | 00000000 _H | 32 | FFFF8610 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA8 | 00000000 _H | 32 | FFFF8614 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC8 | 00000000 _H | 32 | FFFF8618 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC8 | 00000000 _H | 32 | FFFF861C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN8 | 00000000 _H | 32 | FFFF8620 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST8 | 00000000 _H | 32 | FFFF8624 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS8 | 00000000 _H | 32 | FFFF8628 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (130/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DMAC Transfer Status Clear Register | DCSTC8 | 00000000 _H | 32 | FFFF862C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR8 | 00000000 _H | 32 | FFFF8630 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ8 | 00000000 _H | 32 | FFFF8634 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRQC8 | 00000000 _H | 32 | FFFF8638 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA9 | 00000000 _H | 32 | FFFF8640 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA9 | 00000000 _H | 32 | FFFF8644 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC9 | 00000000 _H | 32 | FFFF8648 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT9 | 00000000 _H | 32 | FFFF864C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA9 | 00000000 _H | 32 | FFFF8650 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA9 | 00000000 _H | 32 | FFFF8654 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC9 | 00000000 _H | 32 | FFFF8658 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC9 | 00000000 _H | 32 | FFFF865C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN9 | 00000000 _H | 32 | FFFF8660 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST9 | 00000000 _H | 32 | FFFF8664 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS9 | 00000000 _H | 32 | FFFF8668 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC9 | 00000000 _H | 32 | FFFF866C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR9 | 00000000 _H | 32 | FFFF8670 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ9 | 00000000 _H | 32 | FFFF8674 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRQC9 | 00000000 _H | 32 | FFFF8678 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA10 | 00000000 _H | 32 | FFFF8680 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA10 | 00000000 _H | 32 | FFFF8684 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC10 | 00000000 _H | 32 | FFFF8688 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT10 | 00000000 _H | 32 | FFFF868C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA10 | 00000000 _H | 32 | FFFF8690 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA10 | 00000000 _H | 32 | FFFF8694 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC10 | 00000000 _H | 32 | FFFF8698 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC10 | 00000000 _H | 32 | FFFF869C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN10 | 00000000 _H | 32 | FFFF86A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST10 | 00000000 _H | 32 | FFFF86A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS10 | 00000000 _H | 32 | FFFF86A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC10 | 00000000 _H | 32 | FFFF86AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR10 | 00000000 _H | 32 | FFFF86B0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ10 | 00000000 _H | 32 | FFFF86B4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRQC10 | 00000000 _H | 32 | FFFF86B8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Source Address Register | DSA11 | 00000000 _H | 32 | FFFF86C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Destination Address Register | DDA11 | 00000000 _H | 32 | FFFF86C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Register | DTC11 | 00000000 _H | 32 | FFFF86C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT11 | 00000000 _H | 32 | FFFF86CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA11 | 00000000 _H | 32 | FFFF86D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA11 | 00000000 _H | 32 | FFFF86D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC11 | 00000000 _H | 32 | FFFF86D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC11 | 00000000 _H | 32 | FFFF86DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN11 | 00000000 _H | 32 | FFFF86E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST11 | 00000000 _H | 32 | FFFF86E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS11 | 00000000 _H | 32 | FFFF86E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC11 | 00000000 _H | 32 | FFFF86EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR11 | 00000000 _H | 32 | FFFF86F0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ11 | 00000000 _H | 32 | FFFF86F4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (131/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTFR Transfer Request Clear Register | DTFRQC11 | 00000000 _H | 32 | FFFF86F8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Source Address Register | DSA12 | 00000000 _H | 32 | FFFF8700 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Destination Address Register | DDA12 | 00000000 _H | 32 | FFFF8704 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Count Register | DTC12 | 00000000 _H | 32 | FFFF8708 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Control Register | DTCT12 | 00000000 _H | 32 | FFFF870C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Source Address Register | DRSA12 | 00000000 _H | 32 | FFFF8710 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Destination Address Register | DRDA12 | 00000000 _H | 32 | FFFF8714 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Transfer Count Register | DRTC12 | 00000000 _H | 32 | FFFF8718 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Count Compare Register | DTCC12 | 00000000 _H | 32 | FFFF871C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN12 | 00000000 _H | 32 | FFFF8720 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Register | DCST12 | 00000000 _H | 32 | FFFF8724 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Set Register | DCSTS12 | 00000000 _H | 32 | FFFF8728 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC12 | 00000000 _H | 32 | FFFF872C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Setting Register | DTFR12 | 00000000 _H | 32 | FFFF8730 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Transfer Request Status Register | DTFRQC12 | 00000000 _H | 32 | FFFF8734 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Transfer Request Clear Register | DTFRQC12 | 00000000 _H | 32 | FFFF8738 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Source Address Register | DSA13 | 00000000 _H | 32 | FFFF8740 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Destination Address Register | DDA13 | 00000000 _H | 32 | FFFF8744 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Count Register | DTC13 | 00000000 _H | 32 | FFFF8748 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Control Register | DTCT13 | 00000000 _H | 32 | FFFF874C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Source Address Register | DRSA13 | 00000000 _H | 32 | FFFF8750 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Destination Address Register | DRDA13 | 00000000 _H | 32 | FFFF8754 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Transfer Count Register | DRTC13 | 00000000 _H | 32 | FFFF8758 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Count Compare Register | DTCC13 | 00000000 _H | 32 | FFFF875C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN13 | 00000000 _H | 32 | FFFF8760 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Register | DCST13 | 00000000 _H | 32 | FFFF8764 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Set Register | DCSTS13 | 00000000 _H | 32 | FFFF8768 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC13 | 00000000 _H | 32 | FFFF876C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Setting Register | DTFR13 | 00000000 _H | 32 | FFFF8770 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Transfer Request Status Register | DTFRQC13 | 00000000 _H | 32 | FFFF8774 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Transfer Request Clear Register | DTFRQC13 | 00000000 _H | 32 | FFFF8778 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Source Address Register | DSA14 | 00000000 _H | 32 | FFFF8780 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Destination Address Register | DDA14 | 00000000 _H | 32 | FFFF8784 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Count Register | DTC14 | 00000000 _H | 32 | FFFF8788 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Control Register | DTCT14 | 00000000 _H | 32 | FFFF878C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Source Address Register | DRSA14 | 00000000 _H | 32 | FFFF8790 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Destination Address Register | DRDA14 | 00000000 _H | 32 | FFFF8794 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Reload Transfer Count Register | DRTC14 | 00000000 _H | 32 | FFFF8798 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Count Compare Register | DTCC14 | 00000000 _H | 32 | FFFF879C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN14 | 00000000 _H | 32 | FFFF87A0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Register | DCST14 | 00000000 _H | 32 | FFFF87A4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Set Register | DCSTS14 | 00000000 _H | 32 | FFFF87A8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC14 | 00000000 _H | 32 | FFFF87AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Setting Register | DTFR14 | 00000000 _H | 32 | FFFF87B0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Transfer Request Status Register | DTFRQC14 | 00000000 _H | 32 | FFFF87B4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTFR Transfer Request Clear Register | DTFRQC14 | 00000000 _H | 32 | FFFF87B8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Source Address Register | DSA15 | 00000000 _H | 32 | FFFF87C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DMAC Destination Address Register | DDA15 | 00000000 _H | 32 | FFFF87C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (132/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--|-----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DMAC Transfer Count Register | DTC15 | 00000000 _H | 32 | FFFF87C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Control Register | DTCT15 | 00000000 _H | 32 | FFFF87CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Source Address Register | DRSA15 | 00000000 _H | 32 | FFFF87D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Destination Address Register | DRDA15 | 00000000 _H | 32 | FFFF87D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Reload Transfer Count Register | DRTC15 | 00000000 _H | 32 | FFFF87D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Count Compare Register | DTCC15 | 00000000 _H | 32 | FFFF87DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Channel Operation Enable Setting Register | DCEN15 | 00000000 _H | 32 | FFFF87E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Register | DCST15 | 00000000 _H | 32 | FFFF87E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Set Register | DCSTS15 | 00000000 _H | 32 | FFFF87E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DMAC Transfer Status Clear Register | DCSTC15 | 00000000 _H | 32 | FFFF87EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Setting Register | DTFR15 | 00000000 _H | 32 | FFFF87F0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Status Register | DTFRRQ15 | 00000000 _H | 32 | FFFF87F4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTFR Transfer Request Clear Register | DTFRRQC15 | 00000000 _H | 32 | FFFF87F8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA000 | Undefined | 32 | FFFF9000 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA000 | Undefined | 32 | FFFF9004 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC000 | Undefined | 32 | FFFF9008 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT000 | Undefined | 32 | FFFF900C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA000 | Undefined | 32 | FFFF9010 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA000 | Undefined | 32 | FFFF9014 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC000 | Undefined | 32 | FFFF9018 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC000 | Undefined | 32 | FFFF901C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL000 | 00000000 _H | 32 | FFFF9020 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST000 | 00000000 _H | 32 | FFFF9024 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS000 | 00000000 _H | 32 | FFFF9028 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC000 | 00000000 _H | 32 | FFFF902C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA001 | Undefined | 32 | FFFF9040 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA001 | Undefined | 32 | FFFF9044 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC001 | Undefined | 32 | FFFF9048 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT001 | Undefined | 32 | FFFF904C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA001 | Undefined | 32 | FFFF9050 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA001 | Undefined | 32 | FFFF9054 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC001 | Undefined | 32 | FFFF9058 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC001 | Undefined | 32 | FFFF905C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL001 | 00000000 _H | 32 | FFFF9060 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST001 | 00000000 _H | 32 | FFFF9064 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS001 | 00000000 _H | 32 | FFFF9068 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC001 | 00000000 _H | 32 | FFFF906C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA002 | Undefined | 32 | FFFF9080 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA002 | Undefined | 32 | FFFF9084 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC002 | Undefined | 32 | FFFF9088 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT002 | Undefined | 32 | FFFF908C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA002 | Undefined | 32 | FFFF9090 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA002 | Undefined | 32 | FFFF9094 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC002 | Undefined | 32 | FFFF9098 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC002 | Undefined | 32 | FFFF909C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL002 | 00000000 _H | 32 | FFFF90A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST002 | 00000000 _H | 32 | FFFF90A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS002 | 00000000 _H | 32 | FFFF90A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (133/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC002 | 00000000 _H | 32 | FFFF90AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA003 | Undefined | 32 | FFFF90C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA003 | Undefined | 32 | FFFF90C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC003 | Undefined | 32 | FFFF90C8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT003 | Undefined | 32 | FFFF90CC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA003 | Undefined | 32 | FFFF90D0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA003 | Undefined | 32 | FFFF90D4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC003 | Undefined | 32 | FFFF90D8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC003 | Undefined | 32 | FFFF90DC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL003 | 00000000 _H | 32 | FFFF90E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST003 | 00000000 _H | 32 | FFFF90E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS003 | 00000000 _H | 32 | FFFF90E8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC003 | 00000000 _H | 32 | FFFF90EC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA004 | Undefined | 32 | FFFF9100 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA004 | Undefined | 32 | FFFF9104 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC004 | Undefined | 32 | FFFF9108 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT004 | Undefined | 32 | FFFF910C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA004 | Undefined | 32 | FFFF9110 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA004 | Undefined | 32 | FFFF9114 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC004 | Undefined | 32 | FFFF9118 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC004 | Undefined | 32 | FFFF911C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL004 | 00000000 _H | 32 | FFFF9120 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST004 | 00000000 _H | 32 | FFFF9124 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS004 | 00000000 _H | 32 | FFFF9128 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC004 | 00000000 _H | 32 | FFFF912C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA005 | Undefined | 32 | FFFF9140 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA005 | Undefined | 32 | FFFF9144 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC005 | Undefined | 32 | FFFF9148 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT005 | Undefined | 32 | FFFF914C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA005 | Undefined | 32 | FFFF9150 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA005 | Undefined | 32 | FFFF9154 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC005 | Undefined | 32 | FFFF9158 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC005 | Undefined | 32 | FFFF915C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL005 | 00000000 _H | 32 | FFFF9160 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST005 | 00000000 _H | 32 | FFFF9164 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS005 | 00000000 _H | 32 | FFFF9168 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC005 | 00000000 _H | 32 | FFFF916C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA006 | Undefined | 32 | FFFF9180 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA006 | Undefined | 32 | FFFF9184 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC006 | Undefined | 32 | FFFF9188 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT006 | Undefined | 32 | FFFF918C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA006 | Undefined | 32 | FFFF9190 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA006 | Undefined | 32 | FFFF9194 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC006 | Undefined | 32 | FFFF9198 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC006 | Undefined | 32 | FFFF919C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL006 | 00000000 _H | 32 | FFFF91A0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST006 | 00000000 _H | 32 | FFFF91A4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS006 | 00000000 _H | 32 | FFFF91A8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (134/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC006 | 00000000 _H | 32 | FFFF91AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA007 | Undefined | 32 | FFFF91C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA007 | Undefined | 32 | FFFF91C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC007 | Undefined | 32 | FFFF91C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT007 | Undefined | 32 | FFFF91CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA007 | Undefined | 32 | FFFF91D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA007 | Undefined | 32 | FFFF91D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC007 | Undefined | 32 | FFFF91D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC007 | Undefined | 32 | FFFF91DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL007 | 00000000 _H | 32 | FFFF91E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST007 | 00000000 _H | 32 | FFFF91E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS007 | 00000000 _H | 32 | FFFF91E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC007 | 00000000 _H | 32 | FFFF91EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA008 | Undefined | 32 | FFFF9200 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA008 | Undefined | 32 | FFFF9204 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC008 | Undefined | 32 | FFFF9208 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT008 | Undefined | 32 | FFFF920C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA008 | Undefined | 32 | FFFF9210 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA008 | Undefined | 32 | FFFF9214 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC008 | Undefined | 32 | FFFF9218 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC008 | Undefined | 32 | FFFF921C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL008 | 00000000 _H | 32 | FFFF9220 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST008 | 00000000 _H | 32 | FFFF9224 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS008 | 00000000 _H | 32 | FFFF9228 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC008 | 00000000 _H | 32 | FFFF922C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA009 | Undefined | 32 | FFFF9240 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA009 | Undefined | 32 | FFFF9244 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC009 | Undefined | 32 | FFFF9248 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT009 | Undefined | 32 | FFFF924C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA009 | Undefined | 32 | FFFF9250 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA009 | Undefined | 32 | FFFF9254 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC009 | Undefined | 32 | FFFF9258 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC009 | Undefined | 32 | FFFF925C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL009 | 00000000 _H | 32 | FFFF9260 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST009 | 00000000 _H | 32 | FFFF9264 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS009 | 00000000 _H | 32 | FFFF9268 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC009 | 00000000 _H | 32 | FFFF926C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA010 | Undefined | 32 | FFFF9280 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA010 | Undefined | 32 | FFFF9284 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC010 | Undefined | 32 | FFFF9288 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT010 | Undefined | 32 | FFFF928C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA010 | Undefined | 32 | FFFF9290 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA010 | Undefined | 32 | FFFF9294 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC010 | Undefined | 32 | FFFF9298 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC010 | Undefined | 32 | FFFF929C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL010 | 00000000 _H | 32 | FFFF92A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST010 | 00000000 _H | 32 | FFFF92A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS010 | 00000000 _H | 32 | FFFF92A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (135/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC010 | 00000000 _H | 32 | FFFF92AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA011 | Undefined | 32 | FFFF92C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA011 | Undefined | 32 | FFFF92C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC011 | Undefined | 32 | FFFF92C8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT011 | Undefined | 32 | FFFF92CC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA011 | Undefined | 32 | FFFF92D0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA011 | Undefined | 32 | FFFF92D4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC011 | Undefined | 32 | FFFF92D8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC011 | Undefined | 32 | FFFF92DC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL011 | 00000000 _H | 32 | FFFF92E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST011 | 00000000 _H | 32 | FFFF92E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS011 | 00000000 _H | 32 | FFFF92E8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC011 | 00000000 _H | 32 | FFFF92EC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA012 | Undefined | 32 | FFFF9300 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA012 | Undefined | 32 | FFFF9304 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC012 | Undefined | 32 | FFFF9308 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT012 | Undefined | 32 | FFFF930C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA012 | Undefined | 32 | FFFF9310 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA012 | Undefined | 32 | FFFF9314 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC012 | Undefined | 32 | FFFF9318 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC012 | Undefined | 32 | FFFF931C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL012 | 00000000 _H | 32 | FFFF9320 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST012 | 00000000 _H | 32 | FFFF9324 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS012 | 00000000 _H | 32 | FFFF9328 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC012 | 00000000 _H | 32 | FFFF932C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA013 | Undefined | 32 | FFFF9340 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA013 | Undefined | 32 | FFFF9344 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC013 | Undefined | 32 | FFFF9348 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT013 | Undefined | 32 | FFFF934C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA013 | Undefined | 32 | FFFF9350 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA013 | Undefined | 32 | FFFF9354 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC013 | Undefined | 32 | FFFF9358 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC013 | Undefined | 32 | FFFF935C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL013 | 00000000 _H | 32 | FFFF9360 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST013 | 00000000 _H | 32 | FFFF9364 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS013 | 00000000 _H | 32 | FFFF9368 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC013 | 00000000 _H | 32 | FFFF936C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA014 | Undefined | 32 | FFFF9380 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA014 | Undefined | 32 | FFFF9384 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC014 | Undefined | 32 | FFFF9388 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT014 | Undefined | 32 | FFFF938C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA014 | Undefined | 32 | FFFF9390 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA014 | Undefined | 32 | FFFF9394 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC014 | Undefined | 32 | FFFF9398 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC014 | Undefined | 32 | FFFF939C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL014 | 00000000 _H | 32 | FFFF93A0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST014 | 00000000 _H | 32 | FFFF93A4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS014 | 00000000 _H | 32 | FFFF93A8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (136/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC014 | 00000000 _H | 32 | FFFF93AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA015 | Undefined | 32 | FFFF93C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA015 | Undefined | 32 | FFFF93C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC015 | Undefined | 32 | FFFF93C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT015 | Undefined | 32 | FFFF93CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA015 | Undefined | 32 | FFFF93D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA015 | Undefined | 32 | FFFF93D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC015 | Undefined | 32 | FFFF93D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC015 | Undefined | 32 | FFFF93DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL015 | 00000000 _H | 32 | FFFF93E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST015 | 00000000 _H | 32 | FFFF93E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS015 | 00000000 _H | 32 | FFFF93E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC015 | 00000000 _H | 32 | FFFF93EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA016 | Undefined | 32 | FFFF9400 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA016 | Undefined | 32 | FFFF9404 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC016 | Undefined | 32 | FFFF9408 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT016 | Undefined | 32 | FFFF940C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA016 | Undefined | 32 | FFFF9410 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA016 | Undefined | 32 | FFFF9414 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC016 | Undefined | 32 | FFFF9418 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC016 | Undefined | 32 | FFFF941C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL016 | 00000000 _H | 32 | FFFF9420 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST016 | 00000000 _H | 32 | FFFF9424 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS016 | 00000000 _H | 32 | FFFF9428 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC016 | 00000000 _H | 32 | FFFF942C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA017 | Undefined | 32 | FFFF9440 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA017 | Undefined | 32 | FFFF9444 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC017 | Undefined | 32 | FFFF9448 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT017 | Undefined | 32 | FFFF944C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA017 | Undefined | 32 | FFFF9450 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA017 | Undefined | 32 | FFFF9454 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC017 | Undefined | 32 | FFFF9458 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC017 | Undefined | 32 | FFFF945C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL017 | 00000000 _H | 32 | FFFF9460 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST017 | 00000000 _H | 32 | FFFF9464 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS017 | 00000000 _H | 32 | FFFF9468 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC017 | 00000000 _H | 32 | FFFF946C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA018 | Undefined | 32 | FFFF9480 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA018 | Undefined | 32 | FFFF9484 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC018 | Undefined | 32 | FFFF9488 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT018 | Undefined | 32 | FFFF948C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA018 | Undefined | 32 | FFFF9490 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA018 | Undefined | 32 | FFFF9494 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC018 | Undefined | 32 | FFFF9498 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC018 | Undefined | 32 | FFFF949C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL018 | 00000000 _H | 32 | FFFF94A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST018 | 00000000 _H | 32 | FFFF94A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS018 | 00000000 _H | 32 | FFFF94A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (137/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC018 | 00000000 _H | 32 | FFFF94AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA019 | Undefined | 32 | FFFF94C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA019 | Undefined | 32 | FFFF94C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC019 | Undefined | 32 | FFFF94C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT019 | Undefined | 32 | FFFF94CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA019 | Undefined | 32 | FFFF94D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA019 | Undefined | 32 | FFFF94D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC019 | Undefined | 32 | FFFF94D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC019 | Undefined | 32 | FFFF94DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL019 | 00000000 _H | 32 | FFFF94E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST019 | 00000000 _H | 32 | FFFF94E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS019 | 00000000 _H | 32 | FFFF94E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC019 | 00000000 _H | 32 | FFFF94EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA020 | Undefined | 32 | FFFF9500 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA020 | Undefined | 32 | FFFF9504 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC020 | Undefined | 32 | FFFF9508 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT020 | Undefined | 32 | FFFF950C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA020 | Undefined | 32 | FFFF9510 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA020 | Undefined | 32 | FFFF9514 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC020 | Undefined | 32 | FFFF9518 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC020 | Undefined | 32 | FFFF951C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL020 | 00000000 _H | 32 | FFFF9520 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST020 | 00000000 _H | 32 | FFFF9524 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS020 | 00000000 _H | 32 | FFFF9528 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC020 | 00000000 _H | 32 | FFFF952C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA021 | Undefined | 32 | FFFF9540 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA021 | Undefined | 32 | FFFF9544 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC021 | Undefined | 32 | FFFF9548 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT021 | Undefined | 32 | FFFF954C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA021 | Undefined | 32 | FFFF9550 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA021 | Undefined | 32 | FFFF9554 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC021 | Undefined | 32 | FFFF9558 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC021 | Undefined | 32 | FFFF955C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL021 | 00000000 _H | 32 | FFFF9560 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST021 | 00000000 _H | 32 | FFFF9564 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS021 | 00000000 _H | 32 | FFFF9568 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC021 | 00000000 _H | 32 | FFFF956C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA022 | Undefined | 32 | FFFF9580 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA022 | Undefined | 32 | FFFF9584 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC022 | Undefined | 32 | FFFF9588 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT022 | Undefined | 32 | FFFF958C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA022 | Undefined | 32 | FFFF9590 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA022 | Undefined | 32 | FFFF9594 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC022 | Undefined | 32 | FFFF9598 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC022 | Undefined | 32 | FFFF959C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL022 | 00000000 _H | 32 | FFFF95A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST022 | 00000000 _H | 32 | FFFF95A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS022 | 00000000 _H | 32 | FFFF95A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (138/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC022 | 00000000 _H | 32 | FFFF95AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA023 | Undefined | 32 | FFFF95C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA023 | Undefined | 32 | FFFF95C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC023 | Undefined | 32 | FFFF95C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT023 | Undefined | 32 | FFFF95CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA023 | Undefined | 32 | FFFF95D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA023 | Undefined | 32 | FFFF95D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC023 | Undefined | 32 | FFFF95D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC023 | Undefined | 32 | FFFF95DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL023 | 00000000 _H | 32 | FFFF95E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST023 | 00000000 _H | 32 | FFFF95E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS023 | 00000000 _H | 32 | FFFF95E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC023 | 00000000 _H | 32 | FFFF95EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA024 | Undefined | 32 | FFFF9600 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA024 | Undefined | 32 | FFFF9604 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC024 | Undefined | 32 | FFFF9608 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT024 | Undefined | 32 | FFFF960C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA024 | Undefined | 32 | FFFF9610 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA024 | Undefined | 32 | FFFF9614 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC024 | Undefined | 32 | FFFF9618 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC024 | Undefined | 32 | FFFF961C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL024 | 00000000 _H | 32 | FFFF9620 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST024 | 00000000 _H | 32 | FFFF9624 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS024 | 00000000 _H | 32 | FFFF9628 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC024 | 00000000 _H | 32 | FFFF962C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA025 | Undefined | 32 | FFFF9640 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA025 | Undefined | 32 | FFFF9644 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC025 | Undefined | 32 | FFFF9648 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT025 | Undefined | 32 | FFFF964C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA025 | Undefined | 32 | FFFF9650 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA025 | Undefined | 32 | FFFF9654 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC025 | Undefined | 32 | FFFF9658 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC025 | Undefined | 32 | FFFF965C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL025 | 00000000 _H | 32 | FFFF9660 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST025 | 00000000 _H | 32 | FFFF9664 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS025 | 00000000 _H | 32 | FFFF9668 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC025 | 00000000 _H | 32 | FFFF966C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA026 | Undefined | 32 | FFFF9680 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA026 | Undefined | 32 | FFFF9684 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC026 | Undefined | 32 | FFFF9688 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT026 | Undefined | 32 | FFFF968C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA026 | Undefined | 32 | FFFF9690 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA026 | Undefined | 32 | FFFF9694 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC026 | Undefined | 32 | FFFF9698 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC026 | Undefined | 32 | FFFF969C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL026 | 00000000 _H | 32 | FFFF96A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST026 | 00000000 _H | 32 | FFFF96A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS026 | 00000000 _H | 32 | FFFF96A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (139/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC026 | 00000000 _H | 32 | FFFF96AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA027 | Undefined | 32 | FFFF96C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA027 | Undefined | 32 | FFFF96C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC027 | Undefined | 32 | FFFF96C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT027 | Undefined | 32 | FFFF96CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA027 | Undefined | 32 | FFFF96D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA027 | Undefined | 32 | FFFF96D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC027 | Undefined | 32 | FFFF96D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC027 | Undefined | 32 | FFFF96DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL027 | 00000000 _H | 32 | FFFF96E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST027 | 00000000 _H | 32 | FFFF96E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS027 | 00000000 _H | 32 | FFFF96E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC027 | 00000000 _H | 32 | FFFF96EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA028 | Undefined | 32 | FFFF9700 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA028 | Undefined | 32 | FFFF9704 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC028 | Undefined | 32 | FFFF9708 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT028 | Undefined | 32 | FFFF970C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA028 | Undefined | 32 | FFFF9710 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA028 | Undefined | 32 | FFFF9714 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC028 | Undefined | 32 | FFFF9718 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC028 | Undefined | 32 | FFFF971C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL028 | 00000000 _H | 32 | FFFF9720 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST028 | 00000000 _H | 32 | FFFF9724 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS028 | 00000000 _H | 32 | FFFF9728 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC028 | 00000000 _H | 32 | FFFF972C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA029 | Undefined | 32 | FFFF9740 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA029 | Undefined | 32 | FFFF9744 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC029 | Undefined | 32 | FFFF9748 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT029 | Undefined | 32 | FFFF974C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA029 | Undefined | 32 | FFFF9750 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA029 | Undefined | 32 | FFFF9754 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC029 | Undefined | 32 | FFFF9758 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC029 | Undefined | 32 | FFFF975C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL029 | 00000000 _H | 32 | FFFF9760 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST029 | 00000000 _H | 32 | FFFF9764 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS029 | 00000000 _H | 32 | FFFF9768 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC029 | 00000000 _H | 32 | FFFF976C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA030 | Undefined | 32 | FFFF9780 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA030 | Undefined | 32 | FFFF9784 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC030 | Undefined | 32 | FFFF9788 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT030 | Undefined | 32 | FFFF978C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA030 | Undefined | 32 | FFFF9790 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA030 | Undefined | 32 | FFFF9794 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC030 | Undefined | 32 | FFFF9798 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC030 | Undefined | 32 | FFFF979C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL030 | 00000000 _H | 32 | FFFF97A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST030 | 00000000 _H | 32 | FFFF97A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS030 | 00000000 _H | 32 | FFFF97A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (140/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC030 | 00000000 _H | 32 | FFFF97AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | D TSA031 | Undefined | 32 | FFFF97C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA031 | Undefined | 32 | FFFF97C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC031 | Undefined | 32 | FFFF97C8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT031 | Undefined | 32 | FFFF97CC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | D TRSA031 | Undefined | 32 | FFFF97D0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | D TRDA031 | Undefined | 32 | FFFF97D4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | D TRTC031 | Undefined | 32 | FFFF97D8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC031 | Undefined | 32 | FFFF97DC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL031 | 00000000 _H | 32 | FFFF97E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST031 | 00000000 _H | 32 | FFFF97E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS031 | 00000000 _H | 32 | FFFF97E8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC031 | 00000000 _H | 32 | FFFF97EC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | D TSA032 | Undefined | 32 | FFFF9800 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA032 | Undefined | 32 | FFFF9804 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC032 | Undefined | 32 | FFFF9808 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT032 | Undefined | 32 | FFFF980C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | D TRSA032 | Undefined | 32 | FFFF9810 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | D TRDA032 | Undefined | 32 | FFFF9814 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | D TRTC032 | Undefined | 32 | FFFF9818 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC032 | Undefined | 32 | FFFF981C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL032 | 00000000 _H | 32 | FFFF9820 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST032 | 00000000 _H | 32 | FFFF9824 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS032 | 00000000 _H | 32 | FFFF9828 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC032 | 00000000 _H | 32 | FFFF982C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | D TSA033 | Undefined | 32 | FFFF9840 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA033 | Undefined | 32 | FFFF9844 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC033 | Undefined | 32 | FFFF9848 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT033 | Undefined | 32 | FFFF984C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | D TRSA033 | Undefined | 32 | FFFF9850 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | D TRDA033 | Undefined | 32 | FFFF9854 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | D TRTC033 | Undefined | 32 | FFFF9858 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC033 | Undefined | 32 | FFFF985C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL033 | 00000000 _H | 32 | FFFF9860 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST033 | 00000000 _H | 32 | FFFF9864 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS033 | 00000000 _H | 32 | FFFF9868 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC033 | 00000000 _H | 32 | FFFF986C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | D TSA034 | Undefined | 32 | FFFF9880 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA034 | Undefined | 32 | FFFF9884 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC034 | Undefined | 32 | FFFF9888 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT034 | Undefined | 32 | FFFF988C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | D TRSA034 | Undefined | 32 | FFFF9890 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | D TRDA034 | Undefined | 32 | FFFF9894 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | D TRTC034 | Undefined | 32 | FFFF9898 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC034 | Undefined | 32 | FFFF989C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL034 | 00000000 _H | 32 | FFFF98A0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST034 | 00000000 _H | 32 | FFFF98A4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS034 | 00000000 _H | 32 | FFFF98A8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (141/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC034 | 00000000 _H | 32 | FFFF98AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA035 | Undefined | 32 | FFFF98C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA035 | Undefined | 32 | FFFF98C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC035 | Undefined | 32 | FFFF98C8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT035 | Undefined | 32 | FFFF98CC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA035 | Undefined | 32 | FFFF98D0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA035 | Undefined | 32 | FFFF98D4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC035 | Undefined | 32 | FFFF98D8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC035 | Undefined | 32 | FFFF98DC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL035 | 00000000 _H | 32 | FFFF98E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST035 | 00000000 _H | 32 | FFFF98E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS035 | 00000000 _H | 32 | FFFF98E8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC035 | 00000000 _H | 32 | FFFF98EC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA036 | Undefined | 32 | FFFF9900 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA036 | Undefined | 32 | FFFF9904 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC036 | Undefined | 32 | FFFF9908 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT036 | Undefined | 32 | FFFF990C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA036 | Undefined | 32 | FFFF9910 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA036 | Undefined | 32 | FFFF9914 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC036 | Undefined | 32 | FFFF9918 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC036 | Undefined | 32 | FFFF991C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL036 | 00000000 _H | 32 | FFFF9920 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST036 | 00000000 _H | 32 | FFFF9924 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS036 | 00000000 _H | 32 | FFFF9928 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC036 | 00000000 _H | 32 | FFFF992C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA037 | Undefined | 32 | FFFF9940 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA037 | Undefined | 32 | FFFF9944 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC037 | Undefined | 32 | FFFF9948 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT037 | Undefined | 32 | FFFF994C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA037 | Undefined | 32 | FFFF9950 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA037 | Undefined | 32 | FFFF9954 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC037 | Undefined | 32 | FFFF9958 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC037 | Undefined | 32 | FFFF995C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL037 | 00000000 _H | 32 | FFFF9960 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST037 | 00000000 _H | 32 | FFFF9964 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS037 | 00000000 _H | 32 | FFFF9968 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC037 | 00000000 _H | 32 | FFFF996C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA038 | Undefined | 32 | FFFF9980 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA038 | Undefined | 32 | FFFF9984 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC038 | Undefined | 32 | FFFF9988 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT038 | Undefined | 32 | FFFF998C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA038 | Undefined | 32 | FFFF9990 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA038 | Undefined | 32 | FFFF9994 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC038 | Undefined | 32 | FFFF9998 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC038 | Undefined | 32 | FFFF999C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL038 | 00000000 _H | 32 | FFFF99A0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST038 | 00000000 _H | 32 | FFFF99A4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS038 | 00000000 _H | 32 | FFFF99A8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (142/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC038 | 00000000 _H | 32 | FFFF99AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA039 | Undefined | 32 | FFFF99C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA039 | Undefined | 32 | FFFF99C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC039 | Undefined | 32 | FFFF99C8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT039 | Undefined | 32 | FFFF99CC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA039 | Undefined | 32 | FFFF99D0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA039 | Undefined | 32 | FFFF99D4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC039 | Undefined | 32 | FFFF99D8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC039 | Undefined | 32 | FFFF99DC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL039 | 00000000 _H | 32 | FFFF99E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST039 | 00000000 _H | 32 | FFFF99E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS039 | 00000000 _H | 32 | FFFF99E8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC039 | 00000000 _H | 32 | FFFF99EC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA040 | Undefined | 32 | FFFF9A00 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA040 | Undefined | 32 | FFFF9A04 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC040 | Undefined | 32 | FFFF9A08 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT040 | Undefined | 32 | FFFF9A0C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA040 | Undefined | 32 | FFFF9A10 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA040 | Undefined | 32 | FFFF9A14 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC040 | Undefined | 32 | FFFF9A18 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC040 | Undefined | 32 | FFFF9A1C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL040 | 00000000 _H | 32 | FFFF9A20 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST040 | 00000000 _H | 32 | FFFF9A24 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS040 | 00000000 _H | 32 | FFFF9A28 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC040 | 00000000 _H | 32 | FFFF9A2C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA041 | Undefined | 32 | FFFF9A40 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA041 | Undefined | 32 | FFFF9A44 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC041 | Undefined | 32 | FFFF9A48 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT041 | Undefined | 32 | FFFF9A4C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA041 | Undefined | 32 | FFFF9A50 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA041 | Undefined | 32 | FFFF9A54 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC041 | Undefined | 32 | FFFF9A58 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC041 | Undefined | 32 | FFFF9A5C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL041 | 00000000 _H | 32 | FFFF9A60 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST041 | 00000000 _H | 32 | FFFF9A64 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS041 | 00000000 _H | 32 | FFFF9A68 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC041 | 00000000 _H | 32 | FFFF9A6C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA042 | Undefined | 32 | FFFF9A80 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA042 | Undefined | 32 | FFFF9A84 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC042 | Undefined | 32 | FFFF9A88 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT042 | Undefined | 32 | FFFF9A8C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA042 | Undefined | 32 | FFFF9A90 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA042 | Undefined | 32 | FFFF9A94 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC042 | Undefined | 32 | FFFF9A98 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC042 | Undefined | 32 | FFFF9A9C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL042 | 00000000 _H | 32 | FFFF9AA0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST042 | 00000000 _H | 32 | FFFF9AA4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS042 | 00000000 _H | 32 | FFFF9AA8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (143/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC042 | 00000000 _H | 32 | FFFF9AAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA043 | Undefined | 32 | FFFF9AC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA043 | Undefined | 32 | FFFF9AC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC043 | Undefined | 32 | FFFF9AC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT043 | Undefined | 32 | FFFF9ACC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA043 | Undefined | 32 | FFFF9AD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA043 | Undefined | 32 | FFFF9AD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC043 | Undefined | 32 | FFFF9AD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC043 | Undefined | 32 | FFFF9ADC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL043 | 00000000 _H | 32 | FFFF9AE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST043 | 00000000 _H | 32 | FFFF9AE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS043 | 00000000 _H | 32 | FFFF9AE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC043 | 00000000 _H | 32 | FFFF9AEC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA044 | Undefined | 32 | FFFF9B00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA044 | Undefined | 32 | FFFF9B04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC044 | Undefined | 32 | FFFF9B08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT044 | Undefined | 32 | FFFF9B0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA044 | Undefined | 32 | FFFF9B10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA044 | Undefined | 32 | FFFF9B14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC044 | Undefined | 32 | FFFF9B18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC044 | Undefined | 32 | FFFF9B1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL044 | 00000000 _H | 32 | FFFF9B20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST044 | 00000000 _H | 32 | FFFF9B24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS044 | 00000000 _H | 32 | FFFF9B28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC044 | 00000000 _H | 32 | FFFF9B2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA045 | Undefined | 32 | FFFF9B40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA045 | Undefined | 32 | FFFF9B44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC045 | Undefined | 32 | FFFF9B48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT045 | Undefined | 32 | FFFF9B4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA045 | Undefined | 32 | FFFF9B50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA045 | Undefined | 32 | FFFF9B54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC045 | Undefined | 32 | FFFF9B58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC045 | Undefined | 32 | FFFF9B5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL045 | 00000000 _H | 32 | FFFF9B60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST045 | 00000000 _H | 32 | FFFF9B64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS045 | 00000000 _H | 32 | FFFF9B68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC045 | 00000000 _H | 32 | FFFF9B6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA046 | Undefined | 32 | FFFF9B80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA046 | Undefined | 32 | FFFF9B84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC046 | Undefined | 32 | FFFF9B88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT046 | Undefined | 32 | FFFF9B8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA046 | Undefined | 32 | FFFF9B90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA046 | Undefined | 32 | FFFF9B94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC046 | Undefined | 32 | FFFF9B98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC046 | Undefined | 32 | FFFF9B9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL046 | 00000000 _H | 32 | FFFF9BA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST046 | 00000000 _H | 32 | FFFF9BA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS046 | 00000000 _H | 32 | FFFF9BA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (144/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC046 | 00000000 _H | 32 | FFFF9BAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA047 | Undefined | 32 | FFFF9BC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA047 | Undefined | 32 | FFFF9BC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC047 | Undefined | 32 | FFFF9BC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT047 | Undefined | 32 | FFFF9BCC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA047 | Undefined | 32 | FFFF9BD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA047 | Undefined | 32 | FFFF9BD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC047 | Undefined | 32 | FFFF9BD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC047 | Undefined | 32 | FFFF9BDC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL047 | 00000000 _H | 32 | FFFF9BE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST047 | 00000000 _H | 32 | FFFF9BE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS047 | 00000000 _H | 32 | FFFF9BE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC047 | 00000000 _H | 32 | FFFF9BEC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA048 | Undefined | 32 | FFFF9C00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA048 | Undefined | 32 | FFFF9C04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC048 | Undefined | 32 | FFFF9C08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT048 | Undefined | 32 | FFFF9C0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA048 | Undefined | 32 | FFFF9C10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA048 | Undefined | 32 | FFFF9C14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC048 | Undefined | 32 | FFFF9C18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC048 | Undefined | 32 | FFFF9C1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL048 | 00000000 _H | 32 | FFFF9C20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST048 | 00000000 _H | 32 | FFFF9C24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS048 | 00000000 _H | 32 | FFFF9C28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC048 | 00000000 _H | 32 | FFFF9C2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA049 | Undefined | 32 | FFFF9C40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA049 | Undefined | 32 | FFFF9C44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC049 | Undefined | 32 | FFFF9C48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT049 | Undefined | 32 | FFFF9C4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA049 | Undefined | 32 | FFFF9C50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA049 | Undefined | 32 | FFFF9C54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC049 | Undefined | 32 | FFFF9C58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC049 | Undefined | 32 | FFFF9C5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL049 | 00000000 _H | 32 | FFFF9C60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST049 | 00000000 _H | 32 | FFFF9C64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS049 | 00000000 _H | 32 | FFFF9C68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC049 | 00000000 _H | 32 | FFFF9C6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA050 | Undefined | 32 | FFFF9C80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA050 | Undefined | 32 | FFFF9C84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC050 | Undefined | 32 | FFFF9C88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT050 | Undefined | 32 | FFFF9C8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA050 | Undefined | 32 | FFFF9C90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA050 | Undefined | 32 | FFFF9C94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC050 | Undefined | 32 | FFFF9C98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC050 | Undefined | 32 | FFFF9C9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL050 | 00000000 _H | 32 | FFFF9CA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST050 | 00000000 _H | 32 | FFFF9CA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS050 | 00000000 _H | 32 | FFFF9CA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (145/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC050 | 00000000 _H | 32 | FFFF9CAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA051 | Undefined | 32 | FFFF9CC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA051 | Undefined | 32 | FFFF9CC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC051 | Undefined | 32 | FFFF9CC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT051 | Undefined | 32 | FFFF9CCC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA051 | Undefined | 32 | FFFF9CD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA051 | Undefined | 32 | FFFF9CD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC051 | Undefined | 32 | FFFF9CD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC051 | Undefined | 32 | FFFF9CDC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL051 | 00000000 _H | 32 | FFFF9CE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST051 | 00000000 _H | 32 | FFFF9CE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS051 | 00000000 _H | 32 | FFFF9CE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC051 | 00000000 _H | 32 | FFFF9CEC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA052 | Undefined | 32 | FFFF9D00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA052 | Undefined | 32 | FFFF9D04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC052 | Undefined | 32 | FFFF9D08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT052 | Undefined | 32 | FFFF9D0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA052 | Undefined | 32 | FFFF9D10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA052 | Undefined | 32 | FFFF9D14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC052 | Undefined | 32 | FFFF9D18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC052 | Undefined | 32 | FFFF9D1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL052 | 00000000 _H | 32 | FFFF9D20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST052 | 00000000 _H | 32 | FFFF9D24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS052 | 00000000 _H | 32 | FFFF9D28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC052 | 00000000 _H | 32 | FFFF9D2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA053 | Undefined | 32 | FFFF9D40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA053 | Undefined | 32 | FFFF9D44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC053 | Undefined | 32 | FFFF9D48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT053 | Undefined | 32 | FFFF9D4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA053 | Undefined | 32 | FFFF9D50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA053 | Undefined | 32 | FFFF9D54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC053 | Undefined | 32 | FFFF9D58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC053 | Undefined | 32 | FFFF9D5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL053 | 00000000 _H | 32 | FFFF9D60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST053 | 00000000 _H | 32 | FFFF9D64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS053 | 00000000 _H | 32 | FFFF9D68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC053 | 00000000 _H | 32 | FFFF9D6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA054 | Undefined | 32 | FFFF9D80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA054 | Undefined | 32 | FFFF9D84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC054 | Undefined | 32 | FFFF9D88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT054 | Undefined | 32 | FFFF9D8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA054 | Undefined | 32 | FFFF9D90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA054 | Undefined | 32 | FFFF9D94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC054 | Undefined | 32 | FFFF9D98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC054 | Undefined | 32 | FFFF9D9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL054 | 00000000 _H | 32 | FFFF9DA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST054 | 00000000 _H | 32 | FFFF9DA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS054 | 00000000 _H | 32 | FFFF9DA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (146/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC054 | 00000000 _H | 32 | FFFF9DAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA055 | Undefined | 32 | FFFF9DC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA055 | Undefined | 32 | FFFF9DC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC055 | Undefined | 32 | FFFF9DC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT055 | Undefined | 32 | FFFF9DCC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA055 | Undefined | 32 | FFFF9DD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA055 | Undefined | 32 | FFFF9DD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC055 | Undefined | 32 | FFFF9DD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC055 | Undefined | 32 | FFFF9DDC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL055 | 00000000 _H | 32 | FFFF9DE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST055 | 00000000 _H | 32 | FFFF9DE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS055 | 00000000 _H | 32 | FFFF9DE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC055 | 00000000 _H | 32 | FFFF9DEC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA056 | Undefined | 32 | FFFF9E00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA056 | Undefined | 32 | FFFF9E04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC056 | Undefined | 32 | FFFF9E08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT056 | Undefined | 32 | FFFF9E0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA056 | Undefined | 32 | FFFF9E10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA056 | Undefined | 32 | FFFF9E14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC056 | Undefined | 32 | FFFF9E18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC056 | Undefined | 32 | FFFF9E1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL056 | 00000000 _H | 32 | FFFF9E20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST056 | 00000000 _H | 32 | FFFF9E24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS056 | 00000000 _H | 32 | FFFF9E28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC056 | 00000000 _H | 32 | FFFF9E2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA057 | Undefined | 32 | FFFF9E40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA057 | Undefined | 32 | FFFF9E44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC057 | Undefined | 32 | FFFF9E48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT057 | Undefined | 32 | FFFF9E4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA057 | Undefined | 32 | FFFF9E50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA057 | Undefined | 32 | FFFF9E54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC057 | Undefined | 32 | FFFF9E58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC057 | Undefined | 32 | FFFF9E5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL057 | 00000000 _H | 32 | FFFF9E60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST057 | 00000000 _H | 32 | FFFF9E64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS057 | 00000000 _H | 32 | FFFF9E68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC057 | 00000000 _H | 32 | FFFF9E6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA058 | Undefined | 32 | FFFF9E80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA058 | Undefined | 32 | FFFF9E84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC058 | Undefined | 32 | FFFF9E88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT058 | Undefined | 32 | FFFF9E8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA058 | Undefined | 32 | FFFF9E90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA058 | Undefined | 32 | FFFF9E94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC058 | Undefined | 32 | FFFF9E98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC058 | Undefined | 32 | FFFF9E9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL058 | 00000000 _H | 32 | FFFF9EA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST058 | 00000000 _H | 32 | FFFF9EA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS058 | 00000000 _H | 32 | FFFF9EA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (147/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC058 | 00000000 _H | 32 | FFFF9EAC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA059 | Undefined | 32 | FFFF9EC0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA059 | Undefined | 32 | FFFF9EC4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC059 | Undefined | 32 | FFFF9EC8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT059 | Undefined | 32 | FFFF9ECC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA059 | Undefined | 32 | FFFF9ED0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA059 | Undefined | 32 | FFFF9ED4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC059 | Undefined | 32 | FFFF9ED8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC059 | Undefined | 32 | FFFF9EDC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL059 | 00000000 _H | 32 | FFFF9EE0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST059 | 00000000 _H | 32 | FFFF9EE4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS059 | 00000000 _H | 32 | FFFF9EE8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC059 | 00000000 _H | 32 | FFFF9EEC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA060 | Undefined | 32 | FFFF9F00 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA060 | Undefined | 32 | FFFF9F04 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC060 | Undefined | 32 | FFFF9F08 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT060 | Undefined | 32 | FFFF9F0C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA060 | Undefined | 32 | FFFF9F10 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA060 | Undefined | 32 | FFFF9F14 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC060 | Undefined | 32 | FFFF9F18 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC060 | Undefined | 32 | FFFF9F1C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL060 | 00000000 _H | 32 | FFFF9F20 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST060 | 00000000 _H | 32 | FFFF9F24 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS060 | 00000000 _H | 32 | FFFF9F28 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC060 | 00000000 _H | 32 | FFFF9F2C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA061 | Undefined | 32 | FFFF9F40 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA061 | Undefined | 32 | FFFF9F44 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC061 | Undefined | 32 | FFFF9F48 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT061 | Undefined | 32 | FFFF9F4C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA061 | Undefined | 32 | FFFF9F50 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA061 | Undefined | 32 | FFFF9F54 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC061 | Undefined | 32 | FFFF9F58 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC061 | Undefined | 32 | FFFF9F5C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL061 | 00000000 _H | 32 | FFFF9F60 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST061 | 00000000 _H | 32 | FFFF9F64 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS061 | 00000000 _H | 32 | FFFF9F68 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC061 | 00000000 _H | 32 | FFFF9F6C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA062 | Undefined | 32 | FFFF9F80 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA062 | Undefined | 32 | FFFF9F84 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC062 | Undefined | 32 | FFFF9F88 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT062 | Undefined | 32 | FFFF9F8C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA062 | Undefined | 32 | FFFF9F90 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA062 | Undefined | 32 | FFFF9F94 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC062 | Undefined | 32 | FFFF9F98 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC062 | Undefined | 32 | FFFF9F9C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL062 | 00000000 _H | 32 | FFFF9FA0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST062 | 00000000 _H | 32 | FFFF9FA4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS062 | 00000000 _H | 32 | FFFF9FA8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (148/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC062 | 00000000 _H | 32 | FFFF9FAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA063 | Undefined | 32 | FFFF9FC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA063 | Undefined | 32 | FFFF9FC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC063 | Undefined | 32 | FFFF9FC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT063 | Undefined | 32 | FFFF9FCC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA063 | Undefined | 32 | FFFF9FD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA063 | Undefined | 32 | FFFF9FD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC063 | Undefined | 32 | FFFF9FD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC063 | Undefined | 32 | FFFF9FDC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL063 | 00000000 _H | 32 | FFFF9FE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST063 | 00000000 _H | 32 | FFFF9FE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS063 | 00000000 _H | 32 | FFFF9FE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC063 | 00000000 _H | 32 | FFFF9FEC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA064 | Undefined | 32 | FFFFA000 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA064 | Undefined | 32 | FFFFA004 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC064 | Undefined | 32 | FFFFA008 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT064 | Undefined | 32 | FFFFA00C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA064 | Undefined | 32 | FFFFA010 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA064 | Undefined | 32 | FFFFA014 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC064 | Undefined | 32 | FFFFA018 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC064 | Undefined | 32 | FFFFA01C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL064 | 00000000 _H | 32 | FFFFA020 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST064 | 00000000 _H | 32 | FFFFA024 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS064 | 00000000 _H | 32 | FFFFA028 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC064 | 00000000 _H | 32 | FFFFA02C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA065 | Undefined | 32 | FFFFA040 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA065 | Undefined | 32 | FFFFA044 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC065 | Undefined | 32 | FFFFA048 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT065 | Undefined | 32 | FFFFA04C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA065 | Undefined | 32 | FFFFA050 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA065 | Undefined | 32 | FFFFA054 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC065 | Undefined | 32 | FFFFA058 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC065 | Undefined | 32 | FFFFA05C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL065 | 00000000 _H | 32 | FFFFA060 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST065 | 00000000 _H | 32 | FFFFA064 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS065 | 00000000 _H | 32 | FFFFA068 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC065 | 00000000 _H | 32 | FFFFA06C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA066 | Undefined | 32 | FFFFA080 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA066 | Undefined | 32 | FFFFA084 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC066 | Undefined | 32 | FFFFA088 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT066 | Undefined | 32 | FFFFA08C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA066 | Undefined | 32 | FFFFA090 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA066 | Undefined | 32 | FFFFA094 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC066 | Undefined | 32 | FFFFA098 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC066 | Undefined | 32 | FFFFA09C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL066 | 00000000 _H | 32 | FFFFA0A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST066 | 00000000 _H | 32 | FFFFA0A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS066 | 00000000 _H | 32 | FFFFA0A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (149/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC066 | 00000000 _H | 32 | FFFA0AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA067 | Undefined | 32 | FFFA0C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA067 | Undefined | 32 | FFFA0C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC067 | Undefined | 32 | FFFA0C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT067 | Undefined | 32 | FFFA0CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA067 | Undefined | 32 | FFFA0D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA067 | Undefined | 32 | FFFA0D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC067 | Undefined | 32 | FFFA0D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC067 | Undefined | 32 | FFFA0DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL067 | 00000000 _H | 32 | FFFA0E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST067 | 00000000 _H | 32 | FFFA0E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS067 | 00000000 _H | 32 | FFFA0E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC067 | 00000000 _H | 32 | FFFA0EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA068 | Undefined | 32 | FFFA100 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA068 | Undefined | 32 | FFFA104 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC068 | Undefined | 32 | FFFA108 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT068 | Undefined | 32 | FFFA10C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA068 | Undefined | 32 | FFFA110 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA068 | Undefined | 32 | FFFA114 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC068 | Undefined | 32 | FFFA118 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC068 | Undefined | 32 | FFFA11C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL068 | 00000000 _H | 32 | FFFA120 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST068 | 00000000 _H | 32 | FFFA124 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS068 | 00000000 _H | 32 | FFFA128 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC068 | 00000000 _H | 32 | FFFA12C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA069 | Undefined | 32 | FFFA140 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA069 | Undefined | 32 | FFFA144 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC069 | Undefined | 32 | FFFA148 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT069 | Undefined | 32 | FFFA14C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA069 | Undefined | 32 | FFFA150 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA069 | Undefined | 32 | FFFA154 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC069 | Undefined | 32 | FFFA158 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC069 | Undefined | 32 | FFFA15C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL069 | 00000000 _H | 32 | FFFA160 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST069 | 00000000 _H | 32 | FFFA164 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS069 | 00000000 _H | 32 | FFFA168 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC069 | 00000000 _H | 32 | FFFA16C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA070 | Undefined | 32 | FFFA180 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA070 | Undefined | 32 | FFFA184 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC070 | Undefined | 32 | FFFA188 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT070 | Undefined | 32 | FFFA18C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA070 | Undefined | 32 | FFFA190 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA070 | Undefined | 32 | FFFA194 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC070 | Undefined | 32 | FFFA198 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC070 | Undefined | 32 | FFFA19C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL070 | 00000000 _H | 32 | FFFA1A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST070 | 00000000 _H | 32 | FFFA1A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS070 | 00000000 _H | 32 | FFFA1A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (150/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC070 | 00000000 _H | 32 | FFFA1AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA071 | Undefined | 32 | FFFA1C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA071 | Undefined | 32 | FFFA1C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC071 | Undefined | 32 | FFFA1C8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT071 | Undefined | 32 | FFFA1CC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA071 | Undefined | 32 | FFFA1D0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA071 | Undefined | 32 | FFFA1D4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC071 | Undefined | 32 | FFFA1D8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC071 | Undefined | 32 | FFFA1DC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL071 | 00000000 _H | 32 | FFFA1E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST071 | 00000000 _H | 32 | FFFA1E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS071 | 00000000 _H | 32 | FFFA1E8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC071 | 00000000 _H | 32 | FFFA1EC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA072 | Undefined | 32 | FFFA200 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA072 | Undefined | 32 | FFFA204 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC072 | Undefined | 32 | FFFA208 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT072 | Undefined | 32 | FFFA20C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA072 | Undefined | 32 | FFFA210 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA072 | Undefined | 32 | FFFA214 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC072 | Undefined | 32 | FFFA218 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC072 | Undefined | 32 | FFFA21C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL072 | 00000000 _H | 32 | FFFA220 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST072 | 00000000 _H | 32 | FFFA224 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS072 | 00000000 _H | 32 | FFFA228 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC072 | 00000000 _H | 32 | FFFA22C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA073 | Undefined | 32 | FFFA240 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA073 | Undefined | 32 | FFFA244 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC073 | Undefined | 32 | FFFA248 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT073 | Undefined | 32 | FFFA24C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA073 | Undefined | 32 | FFFA250 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA073 | Undefined | 32 | FFFA254 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC073 | Undefined | 32 | FFFA258 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC073 | Undefined | 32 | FFFA25C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL073 | 00000000 _H | 32 | FFFA260 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST073 | 00000000 _H | 32 | FFFA264 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS073 | 00000000 _H | 32 | FFFA268 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC073 | 00000000 _H | 32 | FFFA26C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA074 | Undefined | 32 | FFFA280 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA074 | Undefined | 32 | FFFA284 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC074 | Undefined | 32 | FFFA288 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT074 | Undefined | 32 | FFFA28C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA074 | Undefined | 32 | FFFA290 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA074 | Undefined | 32 | FFFA294 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC074 | Undefined | 32 | FFFA298 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC074 | Undefined | 32 | FFFA29C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL074 | 00000000 _H | 32 | FFFA2A0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST074 | 00000000 _H | 32 | FFFA2A4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS074 | 00000000 _H | 32 | FFFA2A8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (151/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------|-----------------------|----------------|------------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC074 | 00000000 _H | 32 | FFFA2AC _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Source Address Register | D TSA075 | Undefined | 32 | FFFA2C0 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Destination Address Register | DTDA075 | Undefined | 32 | FFFA2C4 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Register | DTTC075 | Undefined | 32 | FFFA2C8 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Control Register | DTTCT075 | Undefined | 32 | FFFA2CC _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Source Address Register | D TRSA075 | Undefined | 32 | FFFA2D0 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Destination Address Register | D TRDA075 | Undefined | 32 | FFFA2D4 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Transfer Count Register | D TRTC075 | Undefined | 32 | FFFA2D8 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Compare Register | DTTCC075 | Undefined | 32 | FFFA2DC _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Operation Setting Register | DTFSL075 | 00000000 _H | 32 | FFFA2E0 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Status Register | DTFST075 | 00000000 _H | 32 | FFFA2E4 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS075 | 00000000 _H | 32 | FFFA2E8 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC075 | 00000000 _H | 32 | FFFA2EC _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Source Address Register | D TSA076 | Undefined | 32 | FFFA300 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Destination Address Register | DTDA076 | Undefined | 32 | FFFA304 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Register | DTTC076 | Undefined | 32 | FFFA308 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Control Register | DTTCT076 | Undefined | 32 | FFFA30C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Source Address Register | D TRSA076 | Undefined | 32 | FFFA310 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Destination Address Register | D TRDA076 | Undefined | 32 | FFFA314 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Transfer Count Register | D TRTC076 | Undefined | 32 | FFFA318 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Compare Register | DTTCC076 | Undefined | 32 | FFFA31C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Operation Setting Register | DTFSL076 | 00000000 _H | 32 | FFFA320 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Status Register | DTFST076 | 00000000 _H | 32 | FFFA324 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS076 | 00000000 _H | 32 | FFFA328 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC076 | 00000000 _H | 32 | FFFA32C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Source Address Register | D TSA077 | Undefined | 32 | FFFA340 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Destination Address Register | DTDA077 | Undefined | 32 | FFFA344 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Register | DTTC077 | Undefined | 32 | FFFA348 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Control Register | DTTCT077 | Undefined | 32 | FFFA34C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Source Address Register | D TRSA077 | Undefined | 32 | FFFA350 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Destination Address Register | D TRDA077 | Undefined | 32 | FFFA354 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Transfer Count Register | D TRTC077 | Undefined | 32 | FFFA358 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Compare Register | DTTCC077 | Undefined | 32 | FFFA35C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Operation Setting Register | DTFSL077 | 00000000 _H | 32 | FFFA360 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Status Register | DTFST077 | 00000000 _H | 32 | FFFA364 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS077 | 00000000 _H | 32 | FFFA368 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC077 | 00000000 _H | 32 | FFFA36C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Source Address Register | D TSA078 | Undefined | 32 | FFFA380 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Destination Address Register | DTDA078 | Undefined | 32 | FFFA384 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Register | DTTC078 | Undefined | 32 | FFFA388 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Control Register | DTTCT078 | Undefined | 32 | FFFA38C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Source Address Register | D TRSA078 | Undefined | 32 | FFFA390 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Destination Address Register | D TRDA078 | Undefined | 32 | FFFA394 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Reload Transfer Count Register | D TRTC078 | Undefined | 32 | FFFA398 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTS Transfer Count Compare Register | DTTCC078 | Undefined | 32 | FFFA39C _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Operation Setting Register | DTFSL078 | 00000000 _H | 32 | FFFA3A0 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Status Register | DTFST078 | 00000000 _H | 32 | FFFA3A4 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS078 | 00000000 _H | 32 | FFFA3A8 _H 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |

Table A.1 List of Registers (152/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC078 | 00000000 _H | 32 | FFFA3AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA079 | Undefined | 32 | FFFA3C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA079 | Undefined | 32 | FFFA3C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC079 | Undefined | 32 | FFFA3C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT079 | Undefined | 32 | FFFA3CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA079 | Undefined | 32 | FFFA3D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA079 | Undefined | 32 | FFFA3D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC079 | Undefined | 32 | FFFA3D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC079 | Undefined | 32 | FFFA3DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL079 | 00000000 _H | 32 | FFFA3E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST079 | 00000000 _H | 32 | FFFA3E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS079 | 00000000 _H | 32 | FFFA3E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC079 | 00000000 _H | 32 | FFFA3EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA080 | Undefined | 32 | FFFA400 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA080 | Undefined | 32 | FFFA404 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC080 | Undefined | 32 | FFFA408 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT080 | Undefined | 32 | FFFA40C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA080 | Undefined | 32 | FFFA410 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA080 | Undefined | 32 | FFFA414 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC080 | Undefined | 32 | FFFA418 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC080 | Undefined | 32 | FFFA41C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL080 | 00000000 _H | 32 | FFFA420 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST080 | 00000000 _H | 32 | FFFA424 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS080 | 00000000 _H | 32 | FFFA428 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC080 | 00000000 _H | 32 | FFFA42C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA081 | Undefined | 32 | FFFA440 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA081 | Undefined | 32 | FFFA444 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC081 | Undefined | 32 | FFFA448 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT081 | Undefined | 32 | FFFA44C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA081 | Undefined | 32 | FFFA450 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA081 | Undefined | 32 | FFFA454 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC081 | Undefined | 32 | FFFA458 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC081 | Undefined | 32 | FFFA45C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL081 | 00000000 _H | 32 | FFFA460 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST081 | 00000000 _H | 32 | FFFA464 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS081 | 00000000 _H | 32 | FFFA468 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC081 | 00000000 _H | 32 | FFFA46C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA082 | Undefined | 32 | FFFA480 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA082 | Undefined | 32 | FFFA484 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC082 | Undefined | 32 | FFFA488 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT082 | Undefined | 32 | FFFA48C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA082 | Undefined | 32 | FFFA490 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA082 | Undefined | 32 | FFFA494 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC082 | Undefined | 32 | FFFA498 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC082 | Undefined | 32 | FFFA49C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL082 | 00000000 _H | 32 | FFFA4A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST082 | 00000000 _H | 32 | FFFA4A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS082 | 00000000 _H | 32 | FFFA4A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (153/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|-----------|-----------------------|----------------|----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC082 | 00000000 _H | 32 | FFFA4AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA083 | Undefined | 32 | FFFA4C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA083 | Undefined | 32 | FFFA4C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC083 | Undefined | 32 | FFFA4C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT083 | Undefined | 32 | FFFA4CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA083 | Undefined | 32 | FFFA4D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA083 | Undefined | 32 | FFFA4D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC083 | Undefined | 32 | FFFA4D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC083 | Undefined | 32 | FFFA4DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL083 | 00000000 _H | 32 | FFFA4E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST083 | 00000000 _H | 32 | FFFA4E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS083 | 00000000 _H | 32 | FFFA4E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC083 | 00000000 _H | 32 | FFFA4EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA084 | Undefined | 32 | FFFA500 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA084 | Undefined | 32 | FFFA504 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC084 | Undefined | 32 | FFFA508 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT084 | Undefined | 32 | FFFA50C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA084 | Undefined | 32 | FFFA510 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA084 | Undefined | 32 | FFFA514 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC084 | Undefined | 32 | FFFA518 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC084 | Undefined | 32 | FFFA51C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL084 | 00000000 _H | 32 | FFFA520 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST084 | 00000000 _H | 32 | FFFA524 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS084 | 00000000 _H | 32 | FFFA528 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC084 | 00000000 _H | 32 | FFFA52C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA085 | Undefined | 32 | FFFA540 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA085 | Undefined | 32 | FFFA544 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC085 | Undefined | 32 | FFFA548 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT085 | Undefined | 32 | FFFA54C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA085 | Undefined | 32 | FFFA550 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA085 | Undefined | 32 | FFFA554 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC085 | Undefined | 32 | FFFA558 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC085 | Undefined | 32 | FFFA55C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL085 | 00000000 _H | 32 | FFFA560 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST085 | 00000000 _H | 32 | FFFA564 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS085 | 00000000 _H | 32 | FFFA568 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC085 | 00000000 _H | 32 | FFFA56C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA086 | Undefined | 32 | FFFA580 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA086 | Undefined | 32 | FFFA584 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC086 | Undefined | 32 | FFFA588 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT086 | Undefined | 32 | FFFA58C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | D TRSA086 | Undefined | 32 | FFFA590 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | D TRDA086 | Undefined | 32 | FFFA594 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | D TRTC086 | Undefined | 32 | FFFA598 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC086 | Undefined | 32 | FFFA59C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL086 | 00000000 _H | 32 | FFFA5A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST086 | 00000000 _H | 32 | FFFA5A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS086 | 00000000 _H | 32 | FFFA5A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (154/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC086 | 00000000 _H | 32 | FFFA5AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA087 | Undefined | 32 | FFFA5C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA087 | Undefined | 32 | FFFA5C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC087 | Undefined | 32 | FFFA5C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT087 | Undefined | 32 | FFFA5CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA087 | Undefined | 32 | FFFA5D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA087 | Undefined | 32 | FFFA5D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC087 | Undefined | 32 | FFFA5D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC087 | Undefined | 32 | FFFA5DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL087 | 00000000 _H | 32 | FFFA5E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST087 | 00000000 _H | 32 | FFFA5E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS087 | 00000000 _H | 32 | FFFA5E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC087 | 00000000 _H | 32 | FFFA5EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA088 | Undefined | 32 | FFFA600 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA088 | Undefined | 32 | FFFA604 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC088 | Undefined | 32 | FFFA608 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT088 | Undefined | 32 | FFFA60C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA088 | Undefined | 32 | FFFA610 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA088 | Undefined | 32 | FFFA614 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC088 | Undefined | 32 | FFFA618 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC088 | Undefined | 32 | FFFA61C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL088 | 00000000 _H | 32 | FFFA620 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST088 | 00000000 _H | 32 | FFFA624 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS088 | 00000000 _H | 32 | FFFA628 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC088 | 00000000 _H | 32 | FFFA62C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA089 | Undefined | 32 | FFFA640 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA089 | Undefined | 32 | FFFA644 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC089 | Undefined | 32 | FFFA648 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT089 | Undefined | 32 | FFFA64C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA089 | Undefined | 32 | FFFA650 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA089 | Undefined | 32 | FFFA654 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC089 | Undefined | 32 | FFFA658 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC089 | Undefined | 32 | FFFA65C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL089 | 00000000 _H | 32 | FFFA660 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST089 | 00000000 _H | 32 | FFFA664 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS089 | 00000000 _H | 32 | FFFA668 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC089 | 00000000 _H | 32 | FFFA66C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA090 | Undefined | 32 | FFFA680 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA090 | Undefined | 32 | FFFA684 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC090 | Undefined | 32 | FFFA688 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT090 | Undefined | 32 | FFFA68C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA090 | Undefined | 32 | FFFA690 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA090 | Undefined | 32 | FFFA694 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC090 | Undefined | 32 | FFFA698 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC090 | Undefined | 32 | FFFA69C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL090 | 00000000 _H | 32 | FFFA6A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST090 | 00000000 _H | 32 | FFFA6A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS090 | 00000000 _H | 32 | FFFA6A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (155/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC090 | 00000000 _H | 32 | FFFFA6AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA091 | Undefined | 32 | FFFFA6C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA091 | Undefined | 32 | FFFFA6C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC091 | Undefined | 32 | FFFFA6C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT091 | Undefined | 32 | FFFFA6CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA091 | Undefined | 32 | FFFFA6D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA091 | Undefined | 32 | FFFFA6D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC091 | Undefined | 32 | FFFFA6D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC091 | Undefined | 32 | FFFFA6DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL091 | 00000000 _H | 32 | FFFFA6E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST091 | 00000000 _H | 32 | FFFFA6E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS091 | 00000000 _H | 32 | FFFFA6E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC091 | 00000000 _H | 32 | FFFFA6EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA092 | Undefined | 32 | FFFA700 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA092 | Undefined | 32 | FFFA704 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC092 | Undefined | 32 | FFFA708 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT092 | Undefined | 32 | FFFA70C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA092 | Undefined | 32 | FFFA710 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA092 | Undefined | 32 | FFFA714 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC092 | Undefined | 32 | FFFA718 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC092 | Undefined | 32 | FFFA71C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL092 | 00000000 _H | 32 | FFFA720 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST092 | 00000000 _H | 32 | FFFA724 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS092 | 00000000 _H | 32 | FFFA728 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC092 | 00000000 _H | 32 | FFFA72C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA093 | Undefined | 32 | FFFA740 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA093 | Undefined | 32 | FFFA744 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC093 | Undefined | 32 | FFFA748 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT093 | Undefined | 32 | FFFA74C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA093 | Undefined | 32 | FFFA750 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA093 | Undefined | 32 | FFFA754 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC093 | Undefined | 32 | FFFA758 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC093 | Undefined | 32 | FFFA75C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL093 | 00000000 _H | 32 | FFFA760 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST093 | 00000000 _H | 32 | FFFA764 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS093 | 00000000 _H | 32 | FFFA768 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC093 | 00000000 _H | 32 | FFFA76C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA094 | Undefined | 32 | FFFA780 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA094 | Undefined | 32 | FFFA784 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC094 | Undefined | 32 | FFFA788 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT094 | Undefined | 32 | FFFA78C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA094 | Undefined | 32 | FFFA790 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA094 | Undefined | 32 | FFFA794 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC094 | Undefined | 32 | FFFA798 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC094 | Undefined | 32 | FFFA79C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL094 | 00000000 _H | 32 | FFFA7A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST094 | 00000000 _H | 32 | FFFA7A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS094 | 00000000 _H | 32 | FFFA7A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (156/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC094 | 00000000 _H | 32 | FFFFA7AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA095 | Undefined | 32 | FFFFA7C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA095 | Undefined | 32 | FFFFA7C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC095 | Undefined | 32 | FFFFA7C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT095 | Undefined | 32 | FFFFA7CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA095 | Undefined | 32 | FFFFA7D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA095 | Undefined | 32 | FFFFA7D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC095 | Undefined | 32 | FFFFA7D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC095 | Undefined | 32 | FFFFA7DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL095 | 00000000 _H | 32 | FFFFA7E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST095 | 00000000 _H | 32 | FFFFA7E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS095 | 00000000 _H | 32 | FFFFA7E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC095 | 00000000 _H | 32 | FFFFA7EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA096 | Undefined | 32 | FFFFA800 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA096 | Undefined | 32 | FFFFA804 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC096 | Undefined | 32 | FFFFA808 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT096 | Undefined | 32 | FFFFA80C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA096 | Undefined | 32 | FFFFA810 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA096 | Undefined | 32 | FFFFA814 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC096 | Undefined | 32 | FFFFA818 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC096 | Undefined | 32 | FFFFA81C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL096 | 00000000 _H | 32 | FFFFA820 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST096 | 00000000 _H | 32 | FFFFA824 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS096 | 00000000 _H | 32 | FFFFA828 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC096 | 00000000 _H | 32 | FFFFA82C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA097 | Undefined | 32 | FFFFA840 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA097 | Undefined | 32 | FFFFA844 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC097 | Undefined | 32 | FFFFA848 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT097 | Undefined | 32 | FFFFA84C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA097 | Undefined | 32 | FFFFA850 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA097 | Undefined | 32 | FFFFA854 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC097 | Undefined | 32 | FFFFA858 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC097 | Undefined | 32 | FFFFA85C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL097 | 00000000 _H | 32 | FFFFA860 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST097 | 00000000 _H | 32 | FFFFA864 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS097 | 00000000 _H | 32 | FFFFA868 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC097 | 00000000 _H | 32 | FFFFA86C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA098 | Undefined | 32 | FFFFA880 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA098 | Undefined | 32 | FFFFA884 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC098 | Undefined | 32 | FFFFA888 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT098 | Undefined | 32 | FFFFA88C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA098 | Undefined | 32 | FFFFA890 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA098 | Undefined | 32 | FFFFA894 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC098 | Undefined | 32 | FFFFA898 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC098 | Undefined | 32 | FFFFA89C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL098 | 00000000 _H | 32 | FFFFA8A0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST098 | 00000000 _H | 32 | FFFFA8A4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS098 | 00000000 _H | 32 | FFFFA8A8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (157/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC098 | 00000000 _H | 32 | FFFA8AC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA099 | Undefined | 32 | FFFA8C0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA099 | Undefined | 32 | FFFA8C4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC099 | Undefined | 32 | FFFA8C8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT099 | Undefined | 32 | FFFA8CC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA099 | Undefined | 32 | FFFA8D0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA099 | Undefined | 32 | FFFA8D4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC099 | Undefined | 32 | FFFA8D8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC099 | Undefined | 32 | FFFA8DC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL099 | 00000000 _H | 32 | FFFA8E0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST099 | 00000000 _H | 32 | FFFA8E4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS099 | 00000000 _H | 32 | FFFA8E8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC099 | 00000000 _H | 32 | FFFA8EC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA100 | Undefined | 32 | FFFA900 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA100 | Undefined | 32 | FFFA904 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC100 | Undefined | 32 | FFFA908 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT100 | Undefined | 32 | FFFA90C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA100 | Undefined | 32 | FFFA910 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA100 | Undefined | 32 | FFFA914 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC100 | Undefined | 32 | FFFA918 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC100 | Undefined | 32 | FFFA91C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL100 | 00000000 _H | 32 | FFFA920 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST100 | 00000000 _H | 32 | FFFA924 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS100 | 00000000 _H | 32 | FFFA928 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC100 | 00000000 _H | 32 | FFFA92C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA101 | Undefined | 32 | FFFA940 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA101 | Undefined | 32 | FFFA944 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC101 | Undefined | 32 | FFFA948 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT101 | Undefined | 32 | FFFA94C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA101 | Undefined | 32 | FFFA950 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA101 | Undefined | 32 | FFFA954 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC101 | Undefined | 32 | FFFA958 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC101 | Undefined | 32 | FFFA95C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL101 | 00000000 _H | 32 | FFFA960 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST101 | 00000000 _H | 32 | FFFA964 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS101 | 00000000 _H | 32 | FFFA968 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC101 | 00000000 _H | 32 | FFFA96C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | DTSA102 | Undefined | 32 | FFFA980 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA102 | Undefined | 32 | FFFA984 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC102 | Undefined | 32 | FFFA988 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT102 | Undefined | 32 | FFFA98C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA102 | Undefined | 32 | FFFA990 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA102 | Undefined | 32 | FFFA994 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC102 | Undefined | 32 | FFFA998 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC102 | Undefined | 32 | FFFA99C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL102 | 00000000 _H | 32 | FFFA9A0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST102 | 00000000 _H | 32 | FFFA9A4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS102 | 00000000 _H | 32 | FFFA9A8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |

Table A.1 List of Registers (158/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC102 | 00000000 _H | 32 | FFFFA9AC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA103 | Undefined | 32 | FFFFA9C0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA103 | Undefined | 32 | FFFFA9C4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC103 | Undefined | 32 | FFFFA9C8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT103 | Undefined | 32 | FFFFA9CC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA103 | Undefined | 32 | FFFFA9D0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA103 | Undefined | 32 | FFFFA9D4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC103 | Undefined | 32 | FFFFA9D8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC103 | Undefined | 32 | FFFFA9DC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL103 | 00000000 _H | 32 | FFFFA9E0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST103 | 00000000 _H | 32 | FFFFA9E4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS103 | 00000000 _H | 32 | FFFFA9E8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC103 | 00000000 _H | 32 | FFFFA9EC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA104 | Undefined | 32 | FFFFAA00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA104 | Undefined | 32 | FFFFAA04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC104 | Undefined | 32 | FFFFAA08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT104 | Undefined | 32 | FFFFAA0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA104 | Undefined | 32 | FFFFAA10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA104 | Undefined | 32 | FFFFAA14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC104 | Undefined | 32 | FFFFAA18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC104 | Undefined | 32 | FFFFAA1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL104 | 00000000 _H | 32 | FFFFAA20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST104 | 00000000 _H | 32 | FFFFAA24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS104 | 00000000 _H | 32 | FFFFAA28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC104 | 00000000 _H | 32 | FFFFAA2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA105 | Undefined | 32 | FFFFAA40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA105 | Undefined | 32 | FFFFAA44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC105 | Undefined | 32 | FFFFAA48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT105 | Undefined | 32 | FFFFAA4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA105 | Undefined | 32 | FFFFAA50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA105 | Undefined | 32 | FFFFAA54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC105 | Undefined | 32 | FFFFAA58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC105 | Undefined | 32 | FFFFAA5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL105 | 00000000 _H | 32 | FFFFAA60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST105 | 00000000 _H | 32 | FFFFAA64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS105 | 00000000 _H | 32 | FFFFAA68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC105 | 00000000 _H | 32 | FFFFAA6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA106 | Undefined | 32 | FFFFAA80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA106 | Undefined | 32 | FFFFAA84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC106 | Undefined | 32 | FFFFAA88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT106 | Undefined | 32 | FFFFAA8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA106 | Undefined | 32 | FFFFAA90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA106 | Undefined | 32 | FFFFAA94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC106 | Undefined | 32 | FFFFAA98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC106 | Undefined | 32 | FFFFAA9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL106 | 00000000 _H | 32 | FFFFAAA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST106 | 00000000 _H | 32 | FFFFAAA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS106 | 00000000 _H | 32 | FFFFAAA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (159/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC106 | 00000000 _H | 32 | FFFFAAAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA107 | Undefined | 32 | FFFFAAC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA107 | Undefined | 32 | FFFFAAC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC107 | Undefined | 32 | FFFFAAC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT107 | Undefined | 32 | FFFFAAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA107 | Undefined | 32 | FFFFAAD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA107 | Undefined | 32 | FFFFAAD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC107 | Undefined | 32 | FFFFAAD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC107 | Undefined | 32 | FFFFAAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL107 | 00000000 _H | 32 | FFFFAAE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST107 | 00000000 _H | 32 | FFFFAAE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS107 | 00000000 _H | 32 | FFFFAAE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC107 | 00000000 _H | 32 | FFFFAAEC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA108 | Undefined | 32 | FFFFAB00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA108 | Undefined | 32 | FFFFAB04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC108 | Undefined | 32 | FFFFAB08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT108 | Undefined | 32 | FFFFAB0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA108 | Undefined | 32 | FFFFAB10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA108 | Undefined | 32 | FFFFAB14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC108 | Undefined | 32 | FFFFAB18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC108 | Undefined | 32 | FFFFAB1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL108 | 00000000 _H | 32 | FFFFAB20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST108 | 00000000 _H | 32 | FFFFAB24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS108 | 00000000 _H | 32 | FFFFAB28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC108 | 00000000 _H | 32 | FFFFAB2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA109 | Undefined | 32 | FFFFAB40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA109 | Undefined | 32 | FFFFAB44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC109 | Undefined | 32 | FFFFAB48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT109 | Undefined | 32 | FFFFAB4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA109 | Undefined | 32 | FFFFAB50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA109 | Undefined | 32 | FFFFAB54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC109 | Undefined | 32 | FFFFAB58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC109 | Undefined | 32 | FFFFAB5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL109 | 00000000 _H | 32 | FFFFAB60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST109 | 00000000 _H | 32 | FFFFAB64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS109 | 00000000 _H | 32 | FFFFAB68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC109 | 00000000 _H | 32 | FFFFAB6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | D TSA110 | Undefined | 32 | FFFFAB80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA110 | Undefined | 32 | FFFFAB84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC110 | Undefined | 32 | FFFFAB88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT110 | Undefined | 32 | FFFFAB8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA110 | Undefined | 32 | FFFFAB90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA110 | Undefined | 32 | FFFFAB94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC110 | Undefined | 32 | FFFFAB98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC110 | Undefined | 32 | FFFFAB9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL110 | 00000000 _H | 32 | FFFFABA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST110 | 00000000 _H | 32 | FFFFABA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS110 | 00000000 _H | 32 | FFFFABA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (160/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC110 | 00000000 _H | 32 | FFFFABAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA111 | Undefined | 32 | FFFFABC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA111 | Undefined | 32 | FFFFABC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC111 | Undefined | 32 | FFFFABC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT111 | Undefined | 32 | FFFFABC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA111 | Undefined | 32 | FFFFABD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA111 | Undefined | 32 | FFFFABD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC111 | Undefined | 32 | FFFFABD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC111 | Undefined | 32 | FFFFABC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL111 | 00000000 _H | 32 | FFFFABE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST111 | 00000000 _H | 32 | FFFFABE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS111 | 00000000 _H | 32 | FFFFABE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC111 | 00000000 _H | 32 | FFFFABEC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA112 | Undefined | 32 | FFFFAC00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA112 | Undefined | 32 | FFFFAC04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC112 | Undefined | 32 | FFFFAC08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT112 | Undefined | 32 | FFFFAC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA112 | Undefined | 32 | FFFFAC10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA112 | Undefined | 32 | FFFFAC14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC112 | Undefined | 32 | FFFFAC18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC112 | Undefined | 32 | FFFFAC1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL112 | 00000000 _H | 32 | FFFFAC20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST112 | 00000000 _H | 32 | FFFFAC24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS112 | 00000000 _H | 32 | FFFFAC28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC112 | 00000000 _H | 32 | FFFFAC2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA113 | Undefined | 32 | FFFFAC40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA113 | Undefined | 32 | FFFFAC44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC113 | Undefined | 32 | FFFFAC48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT113 | Undefined | 32 | FFFFAC4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA113 | Undefined | 32 | FFFFAC50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA113 | Undefined | 32 | FFFFAC54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC113 | Undefined | 32 | FFFFAC58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC113 | Undefined | 32 | FFFFAC5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL113 | 00000000 _H | 32 | FFFFAC60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST113 | 00000000 _H | 32 | FFFFAC64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS113 | 00000000 _H | 32 | FFFFAC68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC113 | 00000000 _H | 32 | FFFFAC6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA114 | Undefined | 32 | FFFFAC80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA114 | Undefined | 32 | FFFFAC84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC114 | Undefined | 32 | FFFFAC88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT114 | Undefined | 32 | FFFFAC8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA114 | Undefined | 32 | FFFFAC90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA114 | Undefined | 32 | FFFFAC94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC114 | Undefined | 32 | FFFFAC98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC114 | Undefined | 32 | FFFFAC9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL114 | 00000000 _H | 32 | FFFFACA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST114 | 00000000 _H | 32 | FFFFACA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS114 | 00000000 _H | 32 | FFFFACA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (161/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC114 | 00000000 _H | 32 | FFFFACAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA115 | Undefined | 32 | FFFFAC00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA115 | Undefined | 32 | FFFFACC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC115 | Undefined | 32 | FFFFACC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT115 | Undefined | 32 | FFFFACC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA115 | Undefined | 32 | FFFFACD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA115 | Undefined | 32 | FFFFACD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC115 | Undefined | 32 | FFFFACD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC115 | Undefined | 32 | FFFFACDC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL115 | 00000000 _H | 32 | FFFFACE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST115 | 00000000 _H | 32 | FFFFACE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS115 | 00000000 _H | 32 | FFFFACE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC115 | 00000000 _H | 32 | FFFFACE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA116 | Undefined | 32 | FFFFAD00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA116 | Undefined | 32 | FFFFAD04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC116 | Undefined | 32 | FFFFAD08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT116 | Undefined | 32 | FFFFAD0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA116 | Undefined | 32 | FFFFAD10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA116 | Undefined | 32 | FFFFAD14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC116 | Undefined | 32 | FFFFAD18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC116 | Undefined | 32 | FFFFAD1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL116 | 00000000 _H | 32 | FFFFAD20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST116 | 00000000 _H | 32 | FFFFAD24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS116 | 00000000 _H | 32 | FFFFAD28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC116 | 00000000 _H | 32 | FFFFAD2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA117 | Undefined | 32 | FFFFAD40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA117 | Undefined | 32 | FFFFAD44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC117 | Undefined | 32 | FFFFAD48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT117 | Undefined | 32 | FFFFAD4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA117 | Undefined | 32 | FFFFAD50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA117 | Undefined | 32 | FFFFAD54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC117 | Undefined | 32 | FFFFAD58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC117 | Undefined | 32 | FFFFAD5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL117 | 00000000 _H | 32 | FFFFAD60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST117 | 00000000 _H | 32 | FFFFAD64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS117 | 00000000 _H | 32 | FFFFAD68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC117 | 00000000 _H | 32 | FFFFAD6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA118 | Undefined | 32 | FFFFAD80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA118 | Undefined | 32 | FFFFAD84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC118 | Undefined | 32 | FFFFAD88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT118 | Undefined | 32 | FFFFAD8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA118 | Undefined | 32 | FFFFAD90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA118 | Undefined | 32 | FFFFAD94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC118 | Undefined | 32 | FFFFAD98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC118 | Undefined | 32 | FFFFAD9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL118 | 00000000 _H | 32 | FFFFADA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST118 | 00000000 _H | 32 | FFFFADA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS118 | 00000000 _H | 32 | FFFFADA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (162/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|----|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC118 | 00000000 _H | 32 | FFFFADAC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA119 | Undefined | 32 | FFFFADC0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA119 | Undefined | 32 | FFFFADC4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC119 | Undefined | 32 | FFFFADC8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT119 | Undefined | 32 | FFFFADCC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA119 | Undefined | 32 | FFFFADD0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA119 | Undefined | 32 | FFFFADD4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC119 | Undefined | 32 | FFFFADD8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC119 | Undefined | 32 | FFFFADDC _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL119 | 00000000 _H | 32 | FFFFADE0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST119 | 00000000 _H | 32 | FFFFADE4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS119 | 00000000 _H | 32 | FFFFADE8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC119 | 00000000 _H | 32 | FFFFAE0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA120 | Undefined | 32 | FFFFAE00 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA120 | Undefined | 32 | FFFFAE04 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC120 | Undefined | 32 | FFFFAE08 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT120 | Undefined | 32 | FFFFAE0C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA120 | Undefined | 32 | FFFFAE10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA120 | Undefined | 32 | FFFFAE14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC120 | Undefined | 32 | FFFFAE18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC120 | Undefined | 32 | FFFFAE1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL120 | 00000000 _H | 32 | FFFFAE20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST120 | 00000000 _H | 32 | FFFFAE24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS120 | 00000000 _H | 32 | FFFFAE28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC120 | 00000000 _H | 32 | FFFFAE2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA121 | Undefined | 32 | FFFFAE40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA121 | Undefined | 32 | FFFFAE44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC121 | Undefined | 32 | FFFFAE48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT121 | Undefined | 32 | FFFFAE4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA121 | Undefined | 32 | FFFFAE50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA121 | Undefined | 32 | FFFFAE54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC121 | Undefined | 32 | FFFFAE58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC121 | Undefined | 32 | FFFFAE5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL121 | 00000000 _H | 32 | FFFFAE60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST121 | 00000000 _H | 32 | FFFFAE64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS121 | 00000000 _H | 32 | FFFFAE68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC121 | 00000000 _H | 32 | FFFFAE6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Source Address Register | DTSA122 | Undefined | 32 | FFFFAE80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Destination Address Register | DTDA122 | Undefined | 32 | FFFFAE84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Register | DTTC122 | Undefined | 32 | FFFFAE88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Control Register | DTTCT122 | Undefined | 32 | FFFFAE8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Source Address Register | DTRSA122 | Undefined | 32 | FFFFAE90 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Destination Address Register | DTRDA122 | Undefined | 32 | FFFFAE94 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Reload Transfer Count Register | DTRTC122 | Undefined | 32 | FFFFAE98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTS Transfer Count Compare Register | DTTCC122 | Undefined | 32 | FFFFAE9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Operation Setting Register | DTFSL122 | 00000000 _H | 32 | FFFFAEA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST122 | 00000000 _H | 32 | FFFFAEA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS122 | 00000000 _H | 32 | FFFFAEA8 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1 List of Registers (163/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|---|----------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC122 | 00000000 _H | 32 | FFFFAEAC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | D TSA123 | Undefined | 32 | FFFFAE0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA123 | Undefined | 32 | FFFFAE4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC123 | Undefined | 32 | FFFFAE8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT123 | Undefined | 32 | FFFFAECC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA123 | Undefined | 32 | FFFFAED0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA123 | Undefined | 32 | FFFFAED4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC123 | Undefined | 32 | FFFFAED8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC123 | Undefined | 32 | FFFFAEDC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL123 | 00000000 _H | 32 | FFFFAE0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST123 | 00000000 _H | 32 | FFFFAE4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS123 | 00000000 _H | 32 | FFFFAE8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC123 | 00000000 _H | 32 | FFFFAEEC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Source Address Register | D TSA124 | Undefined | 32 | FFFFAF0 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Destination Address Register | DTDA124 | Undefined | 32 | FFFFAF4 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Register | DTTC124 | Undefined | 32 | FFFFAF8 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Control Register | DTTCT124 | Undefined | 32 | FFFFAFCC _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Source Address Register | DTRSA124 | Undefined | 32 | FFFFAF10 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Destination Address Register | DTRDA124 | Undefined | 32 | FFFFAF14 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Reload Transfer Count Register | DTRTC124 | Undefined | 32 | FFFFAF18 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTS Transfer Count Compare Register | DTTCC124 | Undefined | 32 | FFFFAF1C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Operation Setting Register | DTFSL124 | 00000000 _H | 32 | FFFFAF20 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Status Register | DTFST124 | 00000000 _H | 32 | FFFFAF24 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Set Register | DTFSS124 | 00000000 _H | 32 | FFFFAF28 _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| DMASS | DTSFSL Transfer Request Clear Register | DTFSC124 | 00000000 _H | 32 | FFFFAF2C _H | 0 | 32 | √ | √ | √ | √ | √ | √ |
| INTC2 | EI Level Interrupt Control Registers | EIC32 | 008F _H | 16 | FFFFB040 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC33 | 008F _H | 16 | FFFFB042 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC34 | 008F _H | 16 | FFFFB044 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC35 | 008F _H | 16 | FFFFB046 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC36 | 008F _H | 16 | FFFFB048 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC40 | 008F _H | 16 | FFFFB050 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC41 | 008F _H | 16 | FFFFB052 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC42 | 008F _H | 16 | FFFFB054 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC43 | 008F _H | 16 | FFFFB056 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC44 | 008F _H | 16 | FFFFB058 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC45 | 008F _H | 16 | FFFFB05A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC46 | 008F _H | 16 | FFFFB05C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC47 | 008F _H | 16 | FFFFB05E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC48 | 008F _H | 16 | FFFFB060 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC49 | 008F _H | 16 | FFFFB062 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC50 | 008F _H | 16 | FFFFB064 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC51 | 008F _H | 16 | FFFFB066 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (164/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--------------------------------------|--------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Control Registers | EIC52 | 008F _H | 16 | FFFFB068 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC53 | 008F _H | 16 | FFFFB06A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC54 | 008F _H | 16 | FFFFB06C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC55 | 008F _H | 16 | FFFFB06E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC56 | 008F _H | 16 | FFFFB070 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC57 | 008F _H | 16 | FFFFB072 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC58 | 008F _H | 16 | FFFFB074 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC59 | 008F _H | 16 | FFFFB076 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC60 | 008F _H | 16 | FFFFB078 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC61 | 008F _H | 16 | FFFFB07A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC62 | 008F _H | 16 | FFFFB07C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC63 | 008F _H | 16 | FFFFB07E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC64 | 008F _H | 16 | FFFFB080 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC65 | 008F _H | 16 | FFFFB082 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC66 | 008F _H | 16 | FFFFB084 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC67 | 008F _H | 16 | FFFFB086 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC68 | 008F _H | 16 | FFFFB088 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC69 | 008F _H | 16 | FFFFB08A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC70 | 008F _H | 16 | FFFFB08C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC71 | 008F _H | 16 | FFFFB08E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC72 | 008F _H | 16 | FFFFB090 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC73 | 008F _H | 16 | FFFFB092 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC74 | 008F _H | 16 | FFFFB094 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC75 | 008F _H | 16 | FFFFB096 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC76 | 008F _H | 16 | FFFFB098 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC77 | 008F _H | 16 | FFFFB09A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC78 | 008F _H | 16 | FFFFB09C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC79 | 008F _H | 16 | FFFFB09E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC80 | 008F _H | 16 | FFFFB0A0 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC81 | 008F _H | 16 | FFFFB0A2 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC82 | 008F _H | 16 | FFFFB0A4 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC83 | 008F _H | 16 | FFFFB0A6 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC84 | 008F _H | 16 | FFFFB0A8 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC85 | 008F _H | 16 | FFFFB0AA _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC86 | 008F _H | 16 | FFFFB0AC _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC87 | 008F _H | 16 | FFFFB0AE _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |

Table A.1 List of Registers (165/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--------------------------------------|--------|-------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Control Registers | EIC88 | 008F _H | 16 | FFFFB0B0 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC89 | 008F _H | 16 | FFFFB0B2 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC90 | 008F _H | 16 | FFFFB0B4 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC91 | 008F _H | 16 | FFFFB0B6 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC92 | 008F _H | 16 | FFFFB0B8 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC93 | 008F _H | 16 | FFFFB0BA _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC94 | 008F _H | 16 | FFFFB0BC _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC95 | 008F _H | 16 | FFFFB0BE _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC96 | 008F _H | 16 | FFFFB0C0 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC97 | 008F _H | 16 | FFFFB0C2 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC98 | 008F _H | 16 | FFFFB0C4 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC99 | 008F _H | 16 | FFFFB0C6 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC100 | 008F _H | 16 | FFFFB0C8 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC101 | 008F _H | 16 | FFFFB0CA _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC102 | 008F _H | 16 | FFFFB0CC _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC103 | 008F _H | 16 | FFFFB0CE _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC104 | 008F _H | 16 | FFFFB0D0 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC105 | 008F _H | 16 | FFFFB0D2 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC106 | 808F _H | 16 | FFFFB0D4 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC107 | 008F _H | 16 | FFFFB0D6 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC108 | 008F _H | 16 | FFFFB0D8 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC109 | 808F _H | 16 | FFFFB0DA _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC110 | 808F _H | 16 | FFFFB0DC _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC111 | 008F _H | 16 | FFFFB0DE _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC112 | 008F _H | 16 | FFFFB0E0 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC113 | 808F _H | 16 | FFFFB0E2 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC114 | 008F _H | 16 | FFFFB0E4 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC115 | 008F _H | 16 | FFFFB0E6 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC116 | 008F _H | 16 | FFFFB0E8 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC117 | 008F _H | 16 | FFFFB0EA _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC118 | 008F _H | 16 | FFFFB0EC _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC119 | 008F _H | 16 | FFFFB0EE _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC120 | 808F _H | 16 | FFFFB0F0 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC121 | 808F _H | 16 | FFFFB0F2 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC128 | 008F _H | 16 | FFFFB100 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC129 | 008F _H | 16 | FFFFB102 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (166/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--------------------------------------|--------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Control Registers | EIC130 | 008F _H | 16 | FFFFB104 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC131 | 008F _H | 16 | FFFFB106 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC132 | 008F _H | 16 | FFFFB108 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC133 | 008F _H | 16 | FFFFB10A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC134 | 008F _H | 16 | FFFFB10C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC135 | 008F _H | 16 | FFFFB10E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC136 | 008F _H | 16 | FFFFB110 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC137 | 008F _H | 16 | FFFFB112 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC138 | 008F _H | 16 | FFFFB114 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC139 | 008F _H | 16 | FFFFB116 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC140 | 008F _H | 16 | FFFFB118 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC141 | 008F _H | 16 | FFFFB11A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC142 | 008F _H | 16 | FFFFB11C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC143 | 008F _H | 16 | FFFFB11E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC144 | 008F _H | 16 | FFFFB120 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC145 | 008F _H | 16 | FFFFB122 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC146 | 008F _H | 16 | FFFFB124 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC147 | 008F _H | 16 | FFFFB126 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC148 | 008F _H | 16 | FFFFB128 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC149 | 008F _H | 16 | FFFFB12A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC150 | 008F _H | 16 | FFFFB12C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC151 | 008F _H | 16 | FFFFB12E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC152 | 008F _H | 16 | FFFFB130 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC153 | 008F _H | 16 | FFFFB132 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC154 | 008F _H | 16 | FFFFB134 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC155 | 008F _H | 16 | FFFFB136 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC156 | 008F _H | 16 | FFFFB138 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC158 | 008F _H | 16 | FFFFB13C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC159 | 008F _H | 16 | FFFFB13E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC160 | 008F _H | 16 | FFFFB140 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC161 | 008F _H | 16 | FFFFB142 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC162 | 008F _H | 16 | FFFFB144 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC163 | 008F _H | 16 | FFFFB146 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC164 | 008F _H | 16 | FFFFB148 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC165 | 008F _H | 16 | FFFFB14A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC166 | 008F _H | 16 | FFFFB14C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (167/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|--------------------------------------|--------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Control Registers | EIC167 | 008F _H | 16 | FFFFB14E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC168 | 008F _H | 16 | FFFFB150 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC169 | 008F _H | 16 | FFFFB152 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC170 | 008F _H | 16 | FFFFB154 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC171 | 008F _H | 16 | FFFFB156 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC172 | 008F _H | 16 | FFFFB158 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC173 | 008F _H | 16 | FFFFB15A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC174 | 008F _H | 16 | FFFFB15C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC175 | 008F _H | 16 | FFFFB15E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC176 | 008F _H | 16 | FFFFB160 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC177 | 008F _H | 16 | FFFFB162 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC178 | 008F _H | 16 | FFFFB164 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC179 | 008F _H | 16 | FFFFB166 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC180 | 008F _H | 16 | FFFFB168 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC181 | 008F _H | 16 | FFFFB16A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC182 | 008F _H | 16 | FFFFB16C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC183 | 808F _H | 16 | FFFFB16E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC184 | 808F _H | 16 | FFFFB170 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC185 | 808F _H | 16 | FFFFB172 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC186 | 808F _H | 16 | FFFFB174 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC187 | 808F _H | 16 | FFFFB176 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC188 | 808F _H | 16 | FFFFB178 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC189 | 808F _H | 16 | FFFFB17A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC190 | 808F _H | 16 | FFFFB17C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC191 | 808F _H | 16 | FFFFB17E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC192 | 808F _H | 16 | FFFFB180 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC193 | 808F _H | 16 | FFFFB182 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC194 | 808F _H | 16 | FFFFB184 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC195 | 808F _H | 16 | FFFFB186 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC196 | 808F _H | 16 | FFFFB188 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC197 | 808F _H | 16 | FFFFB18A _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC198 | 808F _H | 16 | FFFFB18C _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC199 | 808F _H | 16 | FFFFB18E _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC200 | 808F _H | 16 | FFFFB190 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC201 | 808F _H | 16 | FFFFB192 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC202 | 808F _H | 16 | FFFFB194 _H | 0 | 1, 8, 16 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |

Table A.1 List of Registers (168/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--------------------------------------|--------|-------------------|----------------|-------------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Control Registers | EIC203 | 808F _H | 16 | FFFFB196 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC204 | 808F _H | 16 | FFFFB198 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC218 | 808F _H | 16 | FFFFB1B4 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC219 | 008F _H | 16 | FFFFB1B6 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC220 | 008F _H | 16 | FFFFB1B8 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC221 | 808F _H | 16 | FFFFB1BA _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC226 | 808F _H | 16 | FFFFB1C4 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC227 | 808F _H | 16 | FFFFB1C6 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC228 | 808F _H | 16 | FFFFB1C8 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC229 | 808F _H | 16 | FFFFB1CA _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC230 | 808F _H | 16 | FFFFB1CC _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC231 | 808F _H | 16 | FFFFB1CE _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC232 | 808F _H | 16 | FFFFB1D0 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC233 | 008F _H | 16 | FFFFB1D2 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC234 | 808F _H | 16 | FFFFB1D4 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC235 | 008F _H | 16 | FFFFB1D6 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC236 | 808F _H | 16 | FFFFB1D8 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC237 | 008F _H | 16 | FFFFB1DA _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC238 | 808F _H | 16 | FFFFB1DC _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC239 | 008F _H | 16 | FFFFB1DE _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC240 | 808F _H | 16 | FFFFB1E0 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC241 | 008F _H | 16 | FFFFB1E2 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC242 | 808F _H | 16 | FFFFB1E4 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC243 | 008F _H | 16 | FFFFB1E6 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC244 | 808F _H | 16 | FFFFB1E8 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC245 | 808F _H | 16 | FFFFB1EA _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC246 | 808F _H | 16 | FFFFB1EC _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC247 | 808F _H | 16 | FFFFB1EE _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC248 | 808F _H | 16 | FFFFB1F0 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC249 | 808F _H | 16 | FFFFB1F2 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC250 | 808F _H | 16 | FFFFB1F4 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC251 | 808F _H | 16 | FFFFB1F6 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC252 | 808F _H | 16 | FFFFB1F8 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC253 | 808F _H | 16 | FFFFB1FA _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC254 | 808F _H | 16 | FFFFB1FC _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC256 | 008F _H | 16 | FFFFB200 _H 0 | | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (169/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--------------------------------------|--------|-------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Control Registers | EIC257 | 008F _H | 16 | FFFFB202 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC258 | 008F _H | 16 | FFFFB204 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC259 | 008F _H | 16 | FFFFB206 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC260 | 008F _H | 16 | FFFFB208 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC261 | 008F _H | 16 | FFFFB20A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC262 | 008F _H | 16 | FFFFB20C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC263 | 008F _H | 16 | FFFFB20E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC264 | 008F _H | 16 | FFFFB210 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC265 | 008F _H | 16 | FFFFB212 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC266 | 008F _H | 16 | FFFFB214 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC267 | 008F _H | 16 | FFFFB216 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC268 | 008F _H | 16 | FFFFB218 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC269 | 008F _H | 16 | FFFFB21A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC270 | 008F _H | 16 | FFFFB21C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC271 | 008F _H | 16 | FFFFB21E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC272 | 008F _H | 16 | FFFFB220 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC273 | 008F _H | 16 | FFFFB222 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC274 | 008F _H | 16 | FFFFB224 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC275 | 008F _H | 16 | FFFFB226 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC276 | 008F _H | 16 | FFFFB228 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC277 | 008F _H | 16 | FFFFB22A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC278 | 008F _H | 16 | FFFFB22C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC279 | 008F _H | 16 | FFFFB22E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC280 | 008F _H | 16 | FFFFB230 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC281 | 008F _H | 16 | FFFFB232 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC282 | 008F _H | 16 | FFFFB234 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC283 | 008F _H | 16 | FFFFB236 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC284 | 008F _H | 16 | FFFFB238 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC285 | 008F _H | 16 | FFFFB23A _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC286 | 008F _H | 16 | FFFFB23C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC287 | 008F _H | 16 | FFFFB23E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC288 | 008F _H | 16 | FFFFB240 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC289 | 008F _H | 16 | FFFFB242 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC290 | 008F _H | 16 | FFFFB244 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC291 | 008F _H | 16 | FFFFB246 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC294 | 008F _H | 16 | FFFFB24C _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (170/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|--------------------------------------|--------|----------------------|----------------|-----------------------|---------------------|--------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Control Registers | EIC295 | 008F _H | 16 | FFFFB24E _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC296 | 008F _H | 16 | FFFFB250 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC297 | 008F _H | 16 | FFFFB252 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC298 | 008F _H | 16 | FFFFB254 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC379 | 008F _H | 16 | FFFFB2F6 _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Control Registers | EIC383 | 808F _H | 16 | FFFFB2FE _H | 0 | 1, 8, 16 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR1 | FFFFFFF _H | 32 | FFFFB404 _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR2 | FFFFFFF _H | 32 | FFFFB408 _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR3 | FFFFFFF _H | 32 | FFFFB40C _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR4 | FFFFFFF _H | 32 | FFFFB410 _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR5 | FFFFFFF _H | 32 | FFFFB414 _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR6 | FFFFFFF _H | 32 | FFFFB418 _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR7 | FFFFFFF _H | 32 | FFFFB41C _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR8 | FFFFFFF _H | 32 | FFFFB420 _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR9 | FFFFFFF _H | 32 | FFFFB424 _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Mask Registers | IMR11 | FFFFFFF _H | 32 | FFFFB42C _H | 0 | 1, 8, 16, 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD32 | 0000001 _H | 32 | FFFFB880 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD33 | 0000001 _H | 32 | FFFFB884 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD34 | 0000001 _H | 32 | FFFFB888 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD35 | 0000001 _H | 32 | FFFFB88C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD36 | 0000001 _H | 32 | FFFFB890 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD40 | 0000001 _H | 32 | FFFFB8A0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD41 | 0000001 _H | 32 | FFFFB8A4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD42 | 0000001 _H | 32 | FFFFB8A8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD43 | 0000001 _H | 32 | FFFFB8AC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD44 | 0000001 _H | 32 | FFFFB8B0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD45 | 0000001 _H | 32 | FFFFB8B4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD46 | 0000001 _H | 32 | FFFFB8B8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD47 | 0000001 _H | 32 | FFFFB8BC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD48 | 0000001 _H | 32 | FFFFB8C0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD49 | 0000001 _H | 32 | FFFFB8C4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD50 | 0000001 _H | 32 | FFFFB8C8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD51 | 0000001 _H | 32 | FFFFB8CC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD52 | 0000001 _H | 32 | FFFFB8D0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD53 | 0000001 _H | 32 | FFFFB8D4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD54 | 0000001 _H | 32 | FFFFB8D8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (171/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------------|--------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Bind Registers | EIBD55 | 00000001 _H | 32 | FFFFB8DC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD56 | 00000001 _H | 32 | FFFFB8E0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD57 | 00000001 _H | 32 | FFFFB8E4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD58 | 00000001 _H | 32 | FFFFB8E8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD59 | 00000001 _H | 32 | FFFFB8EC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD60 | 00000001 _H | 32 | FFFFB8F0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD61 | 00000001 _H | 32 | FFFFB8F4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD62 | 00000001 _H | 32 | FFFFB8F8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD63 | 00000001 _H | 32 | FFFFB8FC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD64 | 00000001 _H | 32 | FFFFB900 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD65 | 00000001 _H | 32 | FFFFB904 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD66 | 00000001 _H | 32 | FFFFB908 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD67 | 00000001 _H | 32 | FFFFB90C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD68 | 00000001 _H | 32 | FFFFB910 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD69 | 00000001 _H | 32 | FFFFB914 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD70 | 00000001 _H | 32 | FFFFB918 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD71 | 00000001 _H | 32 | FFFFB91C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD72 | 00000001 _H | 32 | FFFFB920 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD73 | 00000001 _H | 32 | FFFFB924 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD74 | 00000001 _H | 32 | FFFFB928 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD75 | 00000001 _H | 32 | FFFFB92C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD76 | 00000001 _H | 32 | FFFFB930 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD77 | 00000001 _H | 32 | FFFFB934 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD78 | 00000001 _H | 32 | FFFFB938 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD79 | 00000001 _H | 32 | FFFFB93C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD80 | 00000001 _H | 32 | FFFFB940 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD81 | 00000001 _H | 32 | FFFFB944 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD82 | 00000001 _H | 32 | FFFFB948 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD83 | 00000001 _H | 32 | FFFFB94C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD84 | 00000001 _H | 32 | FFFFB950 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD85 | 00000001 _H | 32 | FFFFB954 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD86 | 00000001 _H | 32 | FFFFB958 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD87 | 00000001 _H | 32 | FFFFB95C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD88 | 00000001 _H | 32 | FFFFB960 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD89 | 00000001 _H | 32 | FFFFB964 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD90 | 00000001 _H | 32 | FFFFB968 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (172/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------------|---------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Bind Registers | EIBD91 | 00000001 _H | 32 | FFFFB96C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD92 | 00000001 _H | 32 | FFFFB970 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD93 | 00000001 _H | 32 | FFFFB974 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD94 | 00000001 _H | 32 | FFFFB978 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD95 | 00000001 _H | 32 | FFFFB97C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD96 | 00000001 _H | 32 | FFFFB980 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD97 | 00000001 _H | 32 | FFFFB984 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD98 | 00000001 _H | 32 | FFFFB988 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD99 | 00000001 _H | 32 | FFFFB98C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD100 | 00000001 _H | 32 | FFFFB990 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD101 | 00000001 _H | 32 | FFFFB994 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD102 | 00000001 _H | 32 | FFFFB998 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD103 | 00000001 _H | 32 | FFFFB99C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD104 | 00000001 _H | 32 | FFFFB9A0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD105 | 00000001 _H | 32 | FFFFB9A4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD106 | 00000001 _H | 32 | FFFFB9A8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD107 | 00000001 _H | 32 | FFFFB9AC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD108 | 00000001 _H | 32 | FFFFB9B0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD109 | 00000001 _H | 32 | FFFFB9B4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD110 | 00000001 _H | 32 | FFFFB9B8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD111 | 00000001 _H | 32 | FFFFB9BC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD112 | 00000001 _H | 32 | FFFFB9C0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD113 | 00000001 _H | 32 | FFFFB9C4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD114 | 00000001 _H | 32 | FFFFB9C8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD115 | 00000001 _H | 32 | FFFFB9CC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD116 | 00000001 _H | 32 | FFFFB9D0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD117 | 00000001 _H | 32 | FFFFB9D4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD118 | 00000001 _H | 32 | FFFFB9D8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD119 | 00000001 _H | 32 | FFFFB9DC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD120 | 00000001 _H | 32 | FFFFB9E0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD121 | 00000001 _H | 32 | FFFFB9E4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD128 | 00000001 _H | 32 | FFFFBA00 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD129 | 00000001 _H | 32 | FFFFBA04 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD130 | 00000001 _H | 32 | FFFFBA08 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD131 | 00000001 _H | 32 | FFFFBA0C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD132 | 00000001 _H | 32 | FFFFBA10 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (173/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------------|---------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Bind Registers | EIBD133 | 00000001 _H | 32 | FFFFBA14 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD134 | 00000001 _H | 32 | FFFFBA18 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD135 | 00000001 _H | 32 | FFFFBA1C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD136 | 00000001 _H | 32 | FFFFBA20 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD137 | 00000001 _H | 32 | FFFFBA24 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD138 | 00000001 _H | 32 | FFFFBA28 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD139 | 00000001 _H | 32 | FFFFBA2C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD140 | 00000001 _H | 32 | FFFFBA30 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD141 | 00000001 _H | 32 | FFFFBA34 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD142 | 00000001 _H | 32 | FFFFBA38 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD143 | 00000001 _H | 32 | FFFFBA3C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD144 | 00000001 _H | 32 | FFFFBA40 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD145 | 00000001 _H | 32 | FFFFBA44 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD146 | 00000001 _H | 32 | FFFFBA48 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD147 | 00000001 _H | 32 | FFFFBA4C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD148 | 00000001 _H | 32 | FFFFBA50 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD149 | 00000001 _H | 32 | FFFFBA54 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD150 | 00000001 _H | 32 | FFFFBA58 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD151 | 00000001 _H | 32 | FFFFBA5C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD152 | 00000001 _H | 32 | FFFFBA60 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD153 | 00000001 _H | 32 | FFFFBA64 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD154 | 00000001 _H | 32 | FFFFBA68 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD155 | 00000001 _H | 32 | FFFFBA6C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD156 | 00000001 _H | 32 | FFFFBA70 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD158 | 00000001 _H | 32 | FFFFBA78 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD159 | 00000001 _H | 32 | FFFFBA7C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD160 | 00000001 _H | 32 | FFFFBA80 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD161 | 00000001 _H | 32 | FFFFBA84 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD162 | 00000001 _H | 32 | FFFFBA88 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD163 | 00000001 _H | 32 | FFFFBA8C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD164 | 00000001 _H | 32 | FFFFBA90 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD165 | 00000001 _H | 32 | FFFFBA94 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD166 | 00000001 _H | 32 | FFFFBA98 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD167 | 00000001 _H | 32 | FFFFBA9C _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD168 | 00000001 _H | 32 | FFFFBAA0 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |
| INTC2 | EI Level Interrupt Bind Registers | EIBD169 | 00000001 _H | 32 | FFFFBAA4 _H | 0 | 32 | √ | √ | √ | √ | √ | R:√ W:— |

Table A.1 List of Registers (174/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------------|---------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Bind Registers | EIBD170 | 00000001 _H | 32 | FFFFBAA8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD171 | 00000001 _H | 32 | FFFFBAAC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD172 | 00000001 _H | 32 | FFFFBAB0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD173 | 00000001 _H | 32 | FFFFBAB4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD174 | 00000001 _H | 32 | FFFFBAB8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD175 | 00000001 _H | 32 | FFFFBABC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD176 | 00000001 _H | 32 | FFFFBAC0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD177 | 00000001 _H | 32 | FFFFBAC4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD178 | 00000001 _H | 32 | FFFFBAC8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD179 | 00000001 _H | 32 | FFFFBACC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD180 | 00000001 _H | 32 | FFFFBAD0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD181 | 00000001 _H | 32 | FFFFBAD4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD182 | 00000001 _H | 32 | FFFFBAD8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD183 | 00000001 _H | 32 | FFFFBADC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD184 | 00000001 _H | 32 | FFFFBAE0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD185 | 00000001 _H | 32 | FFFFBAE4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD186 | 00000001 _H | 32 | FFFFBAE8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD187 | 00000001 _H | 32 | FFFFBAEC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD188 | 00000001 _H | 32 | FFFFBAF0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD189 | 00000001 _H | 32 | FFFFBAF4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD190 | 00000001 _H | 32 | FFFFBAF8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD191 | 00000001 _H | 32 | FFFFBAFC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD192 | 00000001 _H | 32 | FFFFBB00 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD193 | 00000001 _H | 32 | FFFFBB04 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD194 | 00000001 _H | 32 | FFFFBB08 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD195 | 00000001 _H | 32 | FFFFBB0C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD196 | 00000001 _H | 32 | FFFFBB10 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD197 | 00000001 _H | 32 | FFFFBB14 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD198 | 00000001 _H | 32 | FFFFBB18 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD199 | 00000001 _H | 32 | FFFFBB1C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD200 | 00000001 _H | 32 | FFFFBB20 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD201 | 00000001 _H | 32 | FFFFBB24 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD202 | 00000001 _H | 32 | FFFFBB28 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD203 | 00000001 _H | 32 | FFFFBB2C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD204 | 00000001 _H | 32 | FFFFBB30 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD218 | 00000001 _H | 32 | FFFFBB68 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (175/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------------|---------|-----------------------|----------------|-----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Bind Registers | EIBD219 | 00000001 _H | 32 | FFFFBB6C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD220 | 00000001 _H | 32 | FFFFBB70 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD221 | 00000001 _H | 32 | FFFFBB74 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD226 | 00000001 _H | 32 | FFFFBB88 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD227 | 00000001 _H | 32 | FFFFBB8C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD228 | 00000001 _H | 32 | FFFFBB90 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD229 | 00000001 _H | 32 | FFFFBB94 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD230 | 00000001 _H | 32 | FFFFBB98 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD231 | 00000001 _H | 32 | FFFFBB9C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD232 | 00000001 _H | 32 | FFFFBBA0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD233 | 00000001 _H | 32 | FFFFBBA4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD234 | 00000001 _H | 32 | FFFFBBA8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD235 | 00000001 _H | 32 | FFFFBBAC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD236 | 00000001 _H | 32 | FFFFBBB0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD237 | 00000001 _H | 32 | FFFFBBB4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD238 | 00000001 _H | 32 | FFFFBBB8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD239 | 00000001 _H | 32 | FFFFBBBC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD240 | 00000001 _H | 32 | FFFFBBC0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD241 | 00000001 _H | 32 | FFFFBBC4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD242 | 00000001 _H | 32 | FFFFBBC8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD243 | 00000001 _H | 32 | FFFFBBCC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD244 | 00000001 _H | 32 | FFFFBBD0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD245 | 00000001 _H | 32 | FFFFBBD4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD246 | 00000001 _H | 32 | FFFFBBD8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD247 | 00000001 _H | 32 | FFFFBDBC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD248 | 00000001 _H | 32 | FFFFBBE0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD249 | 00000001 _H | 32 | FFFFBBE4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD250 | 00000001 _H | 32 | FFFFBBE8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD251 | 00000001 _H | 32 | FFFFBBEC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD252 | 00000001 _H | 32 | FFFFBBF0 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD253 | 00000001 _H | 32 | FFFFBBF4 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD254 | 00000001 _H | 32 | FFFFBBF8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD256 | 00000001 _H | 32 | FFFFBC00 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD257 | 00000001 _H | 32 | FFFFBC04 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD258 | 00000001 _H | 32 | FFFFBC08 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD259 | 00000001 _H | 32 | FFFFBC0C _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |

Table A.1 List of Registers (176/177)

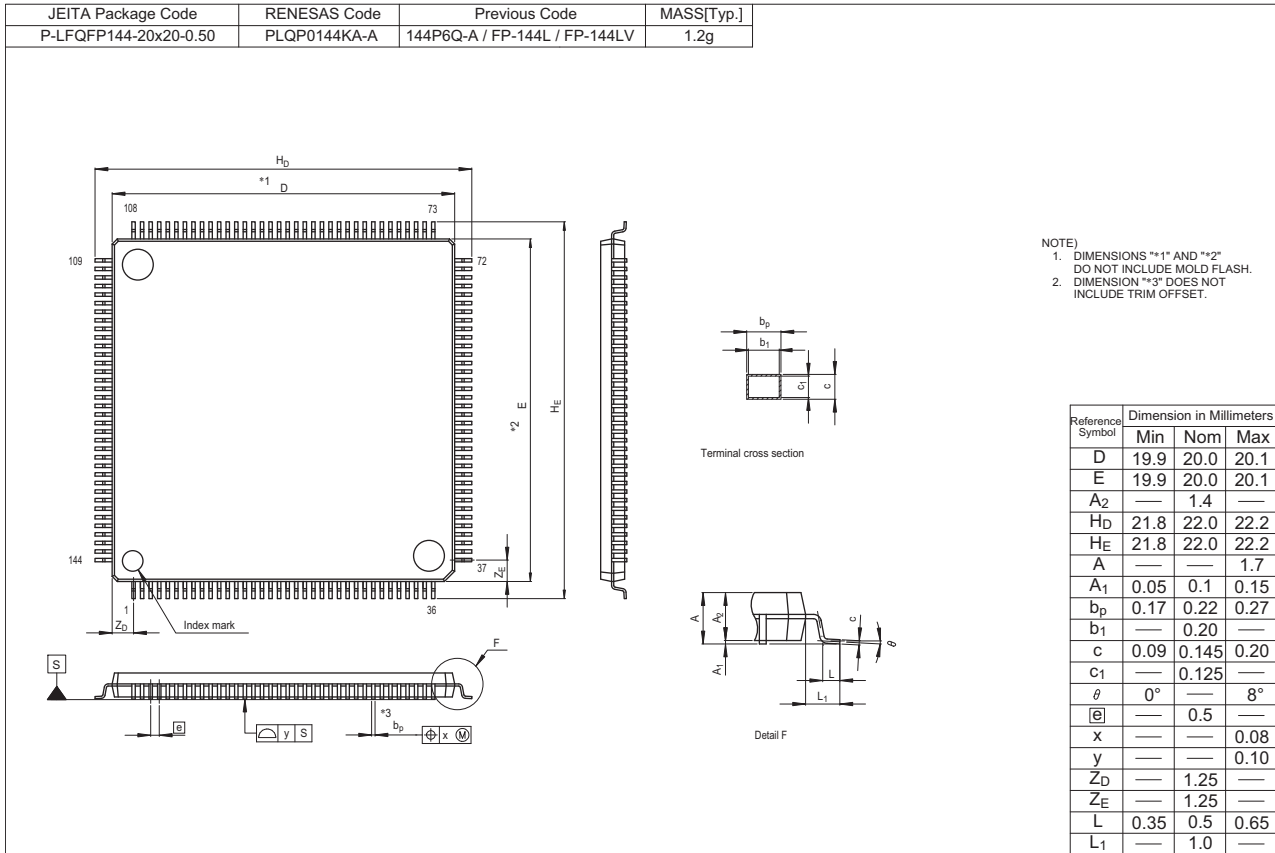
| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (✓: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------------|---------|-----------------------|----------------|-----------------------|---------------------|-------------|---|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Bind Registers | EIBD260 | 00000001 _H | 32 | FFFFBC10 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD261 | 00000001 _H | 32 | FFFFBC14 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD262 | 00000001 _H | 32 | FFFFBC18 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD263 | 00000001 _H | 32 | FFFFBC1C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD264 | 00000001 _H | 32 | FFFFBC20 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD265 | 00000001 _H | 32 | FFFFBC24 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD266 | 00000001 _H | 32 | FFFFBC28 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD267 | 00000001 _H | 32 | FFFFBC2C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD268 | 00000001 _H | 32 | FFFFBC30 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD269 | 00000001 _H | 32 | FFFFBC34 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD270 | 00000001 _H | 32 | FFFFBC38 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD271 | 00000001 _H | 32 | FFFFBC3C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD272 | 00000001 _H | 32 | FFFFBC40 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD273 | 00000001 _H | 32 | FFFFBC44 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD274 | 00000001 _H | 32 | FFFFBC48 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD275 | 00000001 _H | 32 | FFFFBC4C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD276 | 00000001 _H | 32 | FFFFBC50 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD277 | 00000001 _H | 32 | FFFFBC54 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD278 | 00000001 _H | 32 | FFFFBC58 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD279 | 00000001 _H | 32 | FFFFBC5C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD280 | 00000001 _H | 32 | FFFFBC60 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD281 | 00000001 _H | 32 | FFFFBC64 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD282 | 00000001 _H | 32 | FFFFBC68 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD283 | 00000001 _H | 32 | FFFFBC6C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD284 | 00000001 _H | 32 | FFFFBC70 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD285 | 00000001 _H | 32 | FFFFBC74 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD286 | 00000001 _H | 32 | FFFFBC78 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD287 | 00000001 _H | 32 | FFFFBC7C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD288 | 00000001 _H | 32 | FFFFBC80 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD289 | 00000001 _H | 32 | FFFFBC84 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD290 | 00000001 _H | 32 | FFFFBC88 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD291 | 00000001 _H | 32 | FFFFBC8C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD294 | 00000001 _H | 32 | FFFFBC98 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD295 | 00000001 _H | 32 | FFFFBC9C _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD296 | 00000001 _H | 32 | FFFFBCA0 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD297 | 00000001 _H | 32 | FFFFBCA4 _H | 0 | 32 | ✓ | ✓ | ✓ | ✓ | ✓ | R: ✓ W: — |

Table A.1 List of Registers (177/177)

| Module | Register | Symbol | Value after Reset | Number of Bits | Address | Peripheral IP group | Access Size | Reset Source (√: applicable, —: not applicable) | | | | Access Authority | |
|--------|-----------------------------------|---------|-----------------------|----------------|----------------------|---------------------|-------------|--|----------------|----------------|---------------------|------------------|--------------|
| | | | | | | | | Power On Reset | System Reset 1 | System Reset 2 | Application Reset 1 | SV | UM |
| INTC2 | EI Level Interrupt Bind Registers | EIBD298 | 00000001 _H | 32 | FFFBCA8 _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD379 | 00000001 _H | 32 | FFFBDEC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |
| INTC2 | EI Level Interrupt Bind Registers | EIBD383 | 00000001 _H | 32 | FFFBDFC _H | 0 | 32 | √ | √ | √ | √ | √ | R: √ W: — |

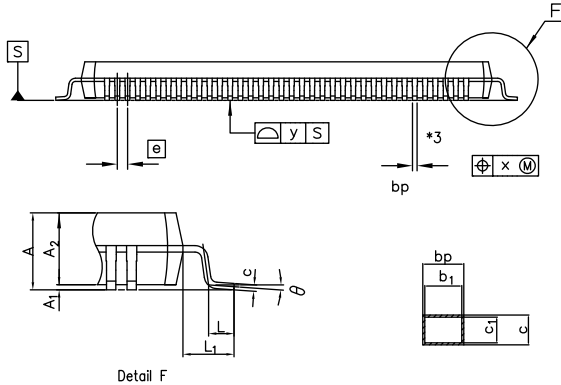
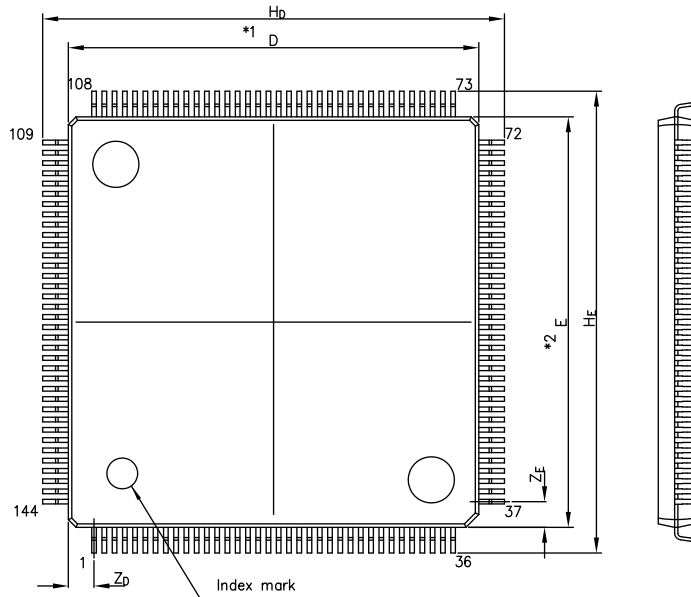
Note: √: applicable, —: not applicable

Appendix B. Package Dimensions



| | | | |
|-----------------------|--------------|---------------|------------|
| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
| P-LFQFP144-16x16-0.40 | PLQP0144LB-A | — | 0.9g |

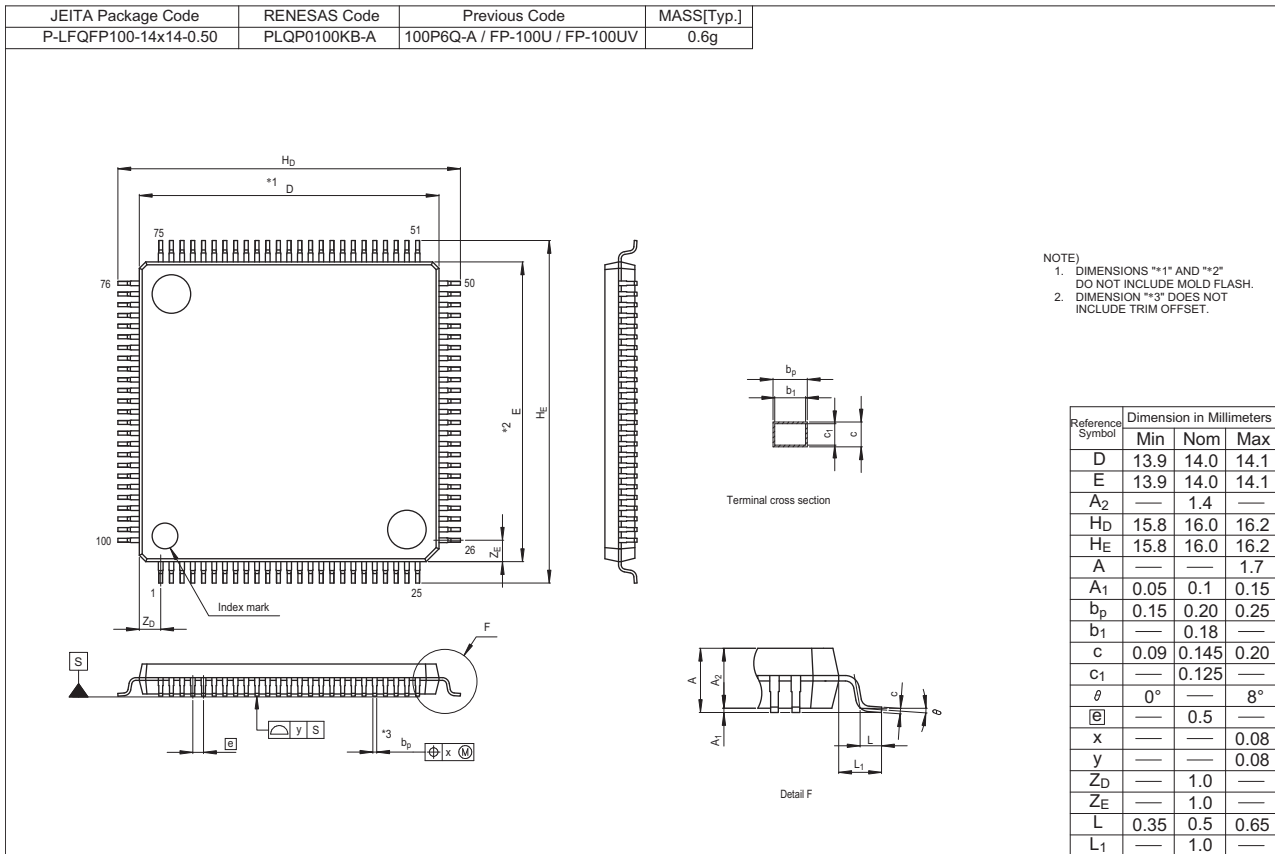
Unit: mm



NOTE)
 1. DIMENSIONS *1* AND *2* DO NOT INCLUDE MOLD FLASH.
 2. DIMENSION *3* DOES NOT INCLUDE TRIM OFFSET.

Terminal cross section

| Reference Symbol | Dimension in Millimeters | | |
|------------------|--------------------------|-------|------|
| | Min | Nom | Max |
| D | 15.9 | 16.0 | 16.1 |
| E | 15.9 | 16.0 | 16.1 |
| A2 | — | 1.4 | — |
| Hb | 17.8 | 18.0 | 18.2 |
| HE | 17.8 | 18.0 | 18.2 |
| A | — | — | 1.7 |
| A1 | 0.05 | 0.10 | 0.15 |
| bp | 0.13 | 0.18 | 0.23 |
| b1 | — | 0.16 | — |
| c | 0.09 | 0.145 | 0.20 |
| c1 | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.4 | — |
| x | — | — | 0.07 |
| y | — | — | 0.08 |
| ZD | — | 1.0 | — |
| ZE | — | 1.0 | — |
| L | 0.35 | 0.5 | 0.65 |
| L1 | — | 1.0 | — |



RH850/P1M-E User's Manual: Hardware

Publication Date: Rev.0.10 Oct 01, 2015
 Rev.1.20 Mar 23, 2018

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351**Renesas Electronics Canada Limited**9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004**Renesas Electronics Europe Limited**Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999**Renesas Electronics Hong Kong Limited**Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022**Renesas Electronics Taiwan Co., Ltd.**13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300**Renesas Electronics Malaysia Sdn.Bhd.**Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510**Renesas Electronics India Pvt. Ltd.**No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777**Renesas Electronics Korea Co., Ltd.**17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

RH850/P1M-E